Extensive-Form Solution Concepts

Extensive-form strategies

When we began our study of game theory with the strategic form, strategy was a primitive of the theory—it was a basic entity with no internal structure. The players each chose a strategy simultaneously, and those choices determined the players' payoffs. In the extensive form of a game, strategies can be very complex. A strategy for a player is a plan for perhaps many actions. Those actions may be spread out in time; between one action for a player and her next, other players' actions may intervene; she might observe those intervening actions perfectly, imperfectly, or not at all.

Game theorists are somewhat schizoid in their interpretation of extensive-form strategies. One popular interpretation is that an extensive-game strategy is a book of instructions which a player could give to a disinterested referee. The player could then go home and have the referee play her desired action whenever called upon to move. (E.g. in the pure-strategy case, there is a page for each of her information sets, on which is written the action to be taken at that information set.) Under this interpretation, the entire plan is not only conceived of before the game is played but, more significantly, a player effectively commits herself—when she files this book of instructions with the referee before the

---

game begins—to particular later actions in response to particular future contingencies.

But at the same time we are supposed to take decision nodes seriously: they mark the occasion at which the *decision* actually takes place (or perhaps better: the occasion at which the player commits to the decision). (In the "play book" interpretation the decision node marks the occasion at which the referee *implements* the previously made decision.) Under this interpretation, a player cannot commit now to a particular decision at a later node. (Any opportunities to commit at an earlier time to a future action can be modeled, but the resulting extensive form would have a decision node at the time of commitment rather than at the time of implementation.)

The utility of the "play book" interpretation is not that it represents how we should think of the game as actually being played. Its role is to specify the level of detail in which we must express a contingent plan in order for it to serve as a well-defined strategy.

Dynamic consistency

In real-world games we look ahead and plan what we would do in various contingencies. However, our plan is not a commitment. As the situation progresses, we constantly reassess our play in response to new information. Even if there is no new information, and the situation is exactly what we forecast it to be, we can still reassess the wisdom of the plan we made earlier. If at some point what we had planned to do no longer seems optimal, we must deviate from our original plan.

Because there are situations in which we cannot commit to a particular decision in the future, a rational player should also be *sequentially rational*: Her planned action at any situation and point in time must actually be optimal at that time and in that situation given her beliefs. We will see that this is a stronger form of rationality assumption than we have made thus far.

Consider the two-player extensive-form game in Figure 1a. Player 1 can choose *U*, ending the game immediately, or *D*, passing the move to player 2, who then chooses *l* or *r*, which ends the game. Figure 1b displays the strategic form of the game.
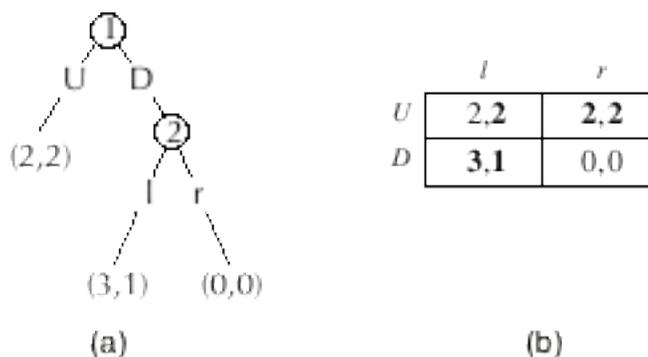

Figure 1: An extensive-form game and its strategic-form representation.

Since we have already developed techniques for analyzing games in strategic form, let's see what we can say about the extensive-form game in Figure 1a by studying its strategic form in Figure 1b. Let's

first ask what outcomes are consistent with common knowledge of rationality—what are the rationalizable outcomes. We note that neither player has a strongly dominated strategy, so the iterated elimination of strongly dominated strategies ends as soon as it begins. Because this is a two-player game and all players' strategies survive this process, every possible outcome is rationalizable.

Let's focus on the outcome in which player 1 chooses U, thus ending the game. Why is U rationalizable for player 1? It is a best response to player 2 choosing r. But that isn't enough. If r is not a rationalizable strategy for player 2, then U being a best response to r wouldn't make U rationalizable for player 1. However, r is a best response to player 1's U. We have therefore created a cycle of beliefs which rationalizes both U for player 1 and r for player 2.

So player 1 chooses U because she believes that player 2 would play r if player 2's node were reached. Note that if player 1 does choose U, then player 2's node is *not* reached. Therefore player 2's planned action at that node is never taken and is completely inconsequential to the payoffs. As you can see from Figure 1b, *any* choice by player 2 is a best response to U because player 2's choice is entirely irrelevant to the payoffs in that case. If play were to reach player 2's node, obviously the only rational thing for player 2 to do would be to choose l and receive 1 rather than the 0 he would receive by playing r. The only reason that player 2's plan to play r at his node is rational is because his beliefs tell him that he will never be called upon to execute this plan.[1]

Player 2's plan to play r would violate his sequential rationality. If player 1 knew that player 2 was sequentially rational, then player 1 could not hold the belief that player 2 would choose r. This belief about 2's choice was the only possible underpinning for player 1's plan to play U. With it removed we see that U is no longer justified for player 1.

To provide ourselves with the analytical power to study arbitrarily complicated extensive-form games we now assume that the sequential rationality of the players is common knowledge.

We have previously seen how games can be decomposed into subgames and how a strategy can be restricted to a subgame. We say that a strategy is rationalizable in a subgame if its restriction to the subgame is rationalizable in the subgame. A strategy is sequentially rationalizable if it is consistent with common knowledge of sequential rationality. In order for a strategy to be sequentially rationalizable, it is clear that it must be rationalizable in every subgame.

In the game of Figure 1a, player 1 must believe that player 2 would play l at his information set if it were reached, because l is the only rationalizable action in that subgame. Player 1 must choose a best response to that belief, therefore the unique sequentially rational outcome of the game is (D, l).

---

[1]   You could feel secure promising to give up all your money on the day it snows in Tucson in July.

Decomposing an extensive form

The subgame

We have previously seen how to isolate a *subgame* from an extensive-form game: it's a subtree which inherits the player set, actions, payoffs, and the player and information partitions from the original game tree and which respects information sets.[2] In order to formalize the procedure of *backwards induction* we used above in the game in Figure 1 we need to also be able to talk about the part of the original game which is excluded from a given subgame; we call this the *difference* game.

Let Γ be an extensive-form game. Because it contains a tree, it contains a set V of vertices and a set E of edges. One node $\mathbb{O} \in V$ is designated to be the initial node and this choice determines the set $Z \subset V$ of terminal nodes and set $X \subset V$ of decision nodes. (We have $V = X \cup Z$, $X \cap Z = \varnothing$, and $\mathbb{O} \in X$.)

Consider any noninitial decision node $x \in X \backslash \{\mathbb{O}\}$ such that Γ can be decomposed into a subgame $\tilde{\Gamma}^x$ whose initial node is x. (As long as the identity of the node x at which the original game is being decomposed is clear from context, I'll suppress the "x" superscript on the $\tilde{\Gamma}$.) The nodes $\tilde{V}$ of the subgame $\tilde{\Gamma}$ are the node x and all of its successors in Γ. The subgame's set $\tilde{V}$ of nodes is partitioned into its decision nodes $\tilde{X}$ and its terminal nodes $\tilde{Z}$, which are defined by restriction of X and Z to $\tilde{V}$. The player set, actions available at each node and the action assignments to each edge, and payoffs to the terminal nodes $\tilde{Z}$ are inherited from the original game by restriction. The set $\tilde{H}$ of subgame information sets is also inherited by restriction from the original game. Because $\tilde{\Gamma}$ is a subgame (rather than just an arbitrary subtree), we know that information sets are respected: $\tilde{H} = \{ h \in H : h \subset \tilde{X} \}$.

For example, in Figure 2, the extensive-form game Γ is decomposed at the player-2 information set α. This node α and all of its successors as well as the edges which represent actions at these nodes form the subtree for the subgame $\tilde{\Gamma}$.

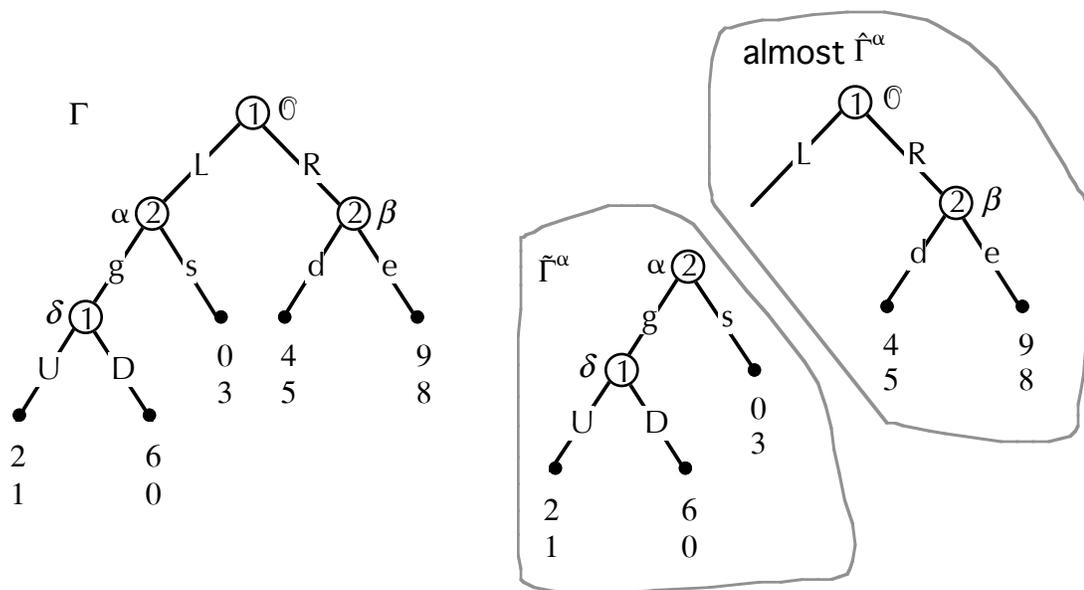[2]   See "Extensive-Form Games," October 5, 1993, pages 13–16.

Figure 2: An extensive-form game Γ decomposed at node α
into the corresponding subgame $\tilde{\Gamma}^\alpha$ and difference game $\hat{\Gamma}^\alpha$.

The difference game

The difference game $\hat{\Gamma}^x$ is partially defined by a set of vertices $\hat{V} \subset V$ and a set of edges $\hat{E} \subset E$. (Again I'll suppress the "x" superscript on the $\tilde{\Gamma}$ when no confusion will result.) The set of vertices $\hat{V}$ contains all of those and only those nodes of the original game which are not in the subgame; i.e. $\hat{V} \supset V \backslash \tilde{V}$ and $\hat{V} \cap \tilde{V} = \varnothing$.[3] (It is because this set is defined in terms of a set-theoretic difference that the residual game is called the difference game.) The decision nodes $\hat{X}$ of the difference game are inherited by restriction from the original game: $\hat{X} = X \cap \hat{V}$. Note in particular that the node x belongs to the subgame $\tilde{\Gamma}$, i.e. $x \in \tilde{V}$, and therefore does not belong to $\hat{V}$. We also note that the original game Γ's set X of decision nodes is partitioned by the decision nodes of the subgame $\tilde{\Gamma}$ and of the difference game $\hat{\Gamma}$; i.e. $X = \tilde{X} \cup \hat{X}$. The difference game's collection $\tilde{H}$ of information sets is defined by restriction: $\hat{H} = \{h \in H : h \subset \hat{X}\}$. The original game's information sets are partitioned by the information sets for the subgame and difference game: $H = \tilde{H} \cup \hat{H}$ and $\tilde{H} \cap \hat{H} = \varnothing$.

Consider the edge $e^x \in E$ of the original game Γ which joined node x to its immediate predecessor, say x', in Γ; i.e. $e^x = (x, x')$.[4] Because node x is not a member of the difference game $\hat{\Gamma}$, in the difference game $\hat{\Gamma}$ the edge e^x is now incomplete: it joins x's immediate predecessor x' on one end but has no node on its other end.[5] (In Figure 2b note that the edge labeled L extending from the initial node $\mathbb{O}$ of the difference game tree has no node on its other end.) Therefore to make the difference game a legitimate extensive form we must augment its set of vertices with an additional terminal node $\hat{z}$ (which we'll soon place where the decision node x had been located). So the difference game's set of nodes is

---

[3]   You'll soon see why I don't more simply just say $\hat{V} = V \backslash \tilde{V}$.

[4]   Recall that an edge is an unordered pair of vertices. (See "Extensive-Form Games," October 5, 1993, page 1.)

[5]   More properly, then, e^x is not an edge in $\hat{\Gamma}$ because it is not an unordered pair of vertices in $\hat{V}$.

$\hat{V} = (V \backslash \tilde{V}) \cup \{\hat{z}\}$ and the set of terminal nodes is $\hat{Z} = (Z \cap \hat{V}) \cup \{\hat{z}\}$.[6] (See Figure 3.)
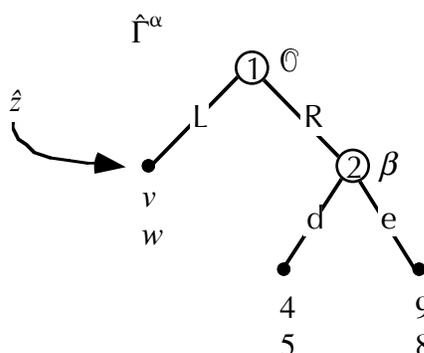

Figure 3: The difference game with artificially added terminal node $\hat{z}$ and redefined edge L connecting $\hat{z}$ to α's immediate predecessor in Γ.

The set of edges $\hat{E}$ of the difference game are defined by first restricting the edges in E to the new set of vertices $\hat{V}$. This however excludes $e^x = (x, x')$, because $x \notin \hat{V}$. So we augment this set of edges with a new one $e' = (x', \hat{z})$ which connects x's immediate predecessor in Γ (viz. x') with the newly added terminal node $\hat{z}$. I.e. $\hat{E} = \{e = (y, z) \in E: y, z \in \hat{V}\} \cup \{e'\}$. The new edge inherits the action assignment of its now-incomplete forbear: $f(e') = f(e^x)$. [In the original game Γ of Figure 2, the edge L was defined by the nodes it directly connected, viz. $L = (\mathbb{0}, \alpha)$. In Figure 3, L has been redefined to be $L = (\mathbb{0}, \hat{z})$.]

Adding the terminal node $\hat{z}$ to the difference tree still doesn't quite succeed in making $\hat{\Gamma}$ a legitimate extensive-form game, because we still do not have a payoff vector assigned to $\hat{z}$. We want to assign $\hat{z}$ the payoff vector which the players would receive if indeed play did proceed to $\hat{z}$. If play did reach $\hat{z}$, that would imply that the subgame $\tilde{\Gamma}$ were reached, so we want to assign to $\hat{z}$ the payoff vector which would be received by the players if they played the subgame $\tilde{\Gamma}$. However, there are typically many different payoff vectors which could be received in the subgame. Before we continue to resolve this indeterminacy, we need to discuss the notion of strategy in the subgame and difference game.

Subgame and difference-game strategies

A strategy profile in an extensive-form game is a map from information sets to actions; i.e. $s: H \to \bar{A}$, where $\bar{A}$ is the set of all actions available at any information set.[7] Given a strategy profile s in the original game Γ, we can break it up into two components: a strategy profile $\tilde{s}$ in the subgame $\tilde{\Gamma}$ and a strategy profile $\hat{s}$ in the difference game $\hat{\Gamma}$. We do this by simply restricting the original strategy profile s to the subsets of information sets $\tilde{H}$ and $\hat{H}$, respectively; $\tilde{s}$ is the restriction of s to $\tilde{H} \subset H$ and $\hat{s}$ is the restriction of s to $\hat{H} \subset H$.[8] Every strategy profile s is equivalent to the pair $(\hat{s}, \tilde{s})$ of difference-game and

---

[6]   This is why I earlier couldn't just say that $\hat{V} = V \backslash \tilde{V}$.

[7]   To be more explicit… For each player $i \in I$, a player-i strategy $s_i: H_i \to \bar{A}_i$ is a map from player-i information sets to player-i actions. Therefore a strategy profile s takes an arbitrary information set $h \in H$, finds the player $\iota(h)$ who owns that information set, and returns the action dictated by $s_{\iota(h)}(h) \in \bar{A}_i \subset \bar{A}$. Therefore the strategy profile is a map from $H \to A$.

[8]   Let $f: X \to Z$ be a function and let $Y \subset X$ be a subset of X. Then we can define a function $\tilde{f}$, the *restriction* of f to Y, as a function whose domain is Y and which agrees with f for all points Y. I.e. $\tilde{f}: Y \to Z$ and $\forall x \in Y, \tilde{f}(x) = f(x)$.

subgame strategy profiles because no information is lost in the decomposition (because $H = \tilde{H} \cap \hat{H}$). In other words we can reconstruct s from the pair $(\hat{s}, \tilde{s})$ according to, for all $h \in H$,

$$s(h) = \begin{cases} \hat{s}(h), & h \in \hat{H}, \\ \tilde{s}(h), & h \in \tilde{H}. \end{cases}$$

We let $\tilde{S}$ and $\hat{S}$ be the space of all subgame and difference-game strategy profiles, respectively. When we restrict a subgame or difference-game strategy profile to player-i's information sets, we define in the natural way player-i subgame and difference-game strategies $\tilde{s}_i \in \tilde{S}_i$ and $\hat{s}_i \in \hat{S}_i$ and the corresponding player-i strategy spaces $\tilde{S}_i$ and $\hat{S}_i$. We get the same equivalence between player-i strategies in the original game and pairs of strategies in the difference-game and subgame as we did between strategy profiles; i.e. every player-i strategy profile $s_i \in S_i$ in the original game determines a pair $(\hat{s}_i; \tilde{s}_i)$ of player-i difference-game and subgame strategies, and vice versa.

Consider a strategy profile $(t_i, s_{-i}) \in S$ of the original game in which player i chooses $t_i \in S_i$ while her opponents choose the deleted strategy profile $s_{-i} \in S_{-i}$. We can decompose each player's strategy into an individual difference-game and subgame strategy; i.e. $t_i = (\hat{t}_i; \tilde{t}_i)$ and $s_{-i} = (\hat{s}_{-i}; \tilde{s}_{-i})$.[9] You can show that the following equivalence holds:

$$(t_i, s_{-i}) = ((\hat{t}_i; \tilde{t}_i), (\hat{s}_{-i}, \tilde{s}_{-i})) = ((\hat{t}_i, \hat{s}_{-i}); (\tilde{t}_i, \tilde{s}_{-i})). \tag{1}$$

For an example of the decomposition of strategy profiles, reconsider Figure 2. We can write a strategy profile in the original game Γ in the form

$$s = (s_1(\mathbb{O}), s_1(\delta); s_2(\alpha), s_2(\beta)).$$

When we restrict $s \in S$ to the subgame and to the difference game, respectively, we obtain the strategy profiles $\tilde{s} \in \tilde{S}$ and $\hat{s} \in \hat{S}$, which we can write in the form

$$\tilde{s} = (s_1(\delta); s_2(\alpha)),$$

$$\hat{s} = (s_1(\mathbb{O}); s_2(\beta)).$$

(Note that the pair $(\hat{s}, \tilde{s})$ contains all the information present in s.) For example, see Figure 4. Here I have shown the added terminal node $\hat{z}$ and its associated payoff vector $(v, w) \in \mathbb{R}^2$, whose values have yet to be determined. The strategy profile indicated in the original game Γ is $s = (L, U; s, e)$. When restricted to the subgame it becomes $\tilde{s} = (U; s)$. When restricted to the difference game, it becomes $\hat{s} = (L; e)$.

---

[9]   Note the distinction between the use of a comma and a semicolon in (t_i, s_{-i}) and $(\hat{t}_i; \tilde{t}_i)$, respectively.
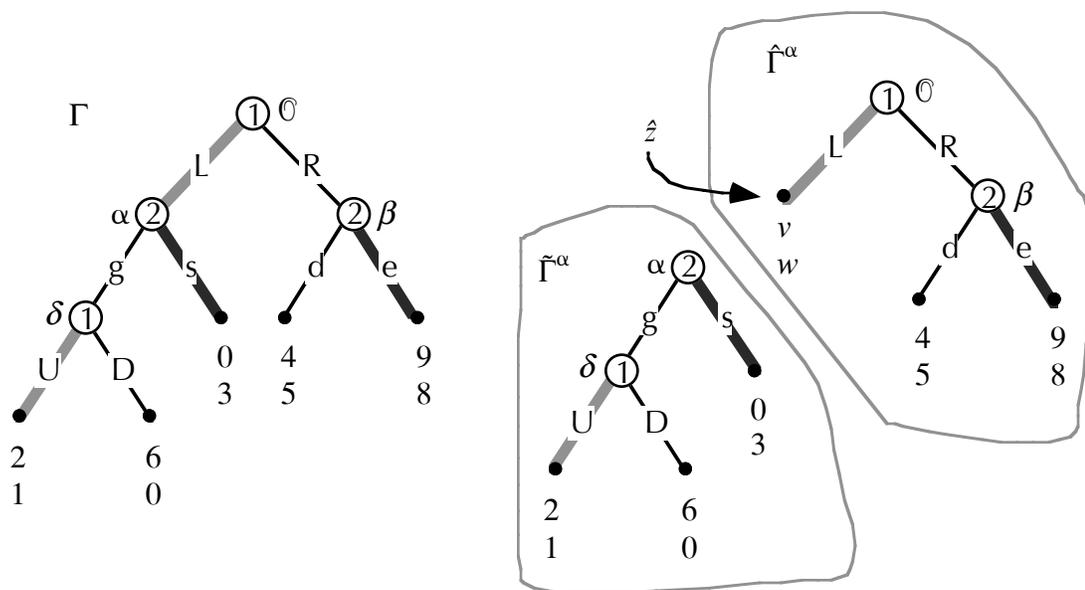
Figure 4: A strategy profile in the game Γ decomposed into strategy profiles
for the subgame and difference games.

A subgame strategy $\tilde{s}$ determines a particular difference game

Now we can address the question of what payoff vector to assign to the artificially added terminal node $\hat{z}$. Every strategy profile $\tilde{s} \in \tilde{S}$ of the subgame determines a terminal node and therefore a payoff vector in the subgame. If we believe that the subgame would be played according to some subgame strategy profile $\tilde{s}$ if that subgame were reached, then we would want to award the newly added terminal node $\hat{z}$ in the difference game the payoff vector of the subgame which would be received if the subgame were reached and $\tilde{s}$ were played there. Each choice of subgame strategy profile $\tilde{s} \in \tilde{S}$ determines such a payoff and therefore completes the determination of a difference game $\hat{\Gamma}(\tilde{s})$. Formally we say that $\hat{\mu}(\hat{z}) = \tilde{u}(\tilde{s})$, where $\hat{\mu} : \hat{Z} \to \mathbb{R}^n$ is the function assigning utilities to outcomes in the difference game. For example, in Figure 4 the specification of the subgame strategy profile $\tilde{s} = (U; s)$ determines the payoff vector $(0, 3)$ in the subgame and therefore determines the payoff vector for $\hat{z}$, viz. $(v, w) = (0, 3)$, in the difference game $\hat{\Gamma}((U; s))$. See Figure 5.

Payoffs in the original game, the subgame, and the difference game

We let, as usual, $u : S \to \mathbb{R}^n$ be the payoff-vector function in the game Γ. We let $\tilde{u} : \tilde{S} \to \mathbb{R}^n$ be the payoff-vector function in the subgame $\tilde{\Gamma}$; i.e. $\tilde{u}(\tilde{s}) \in \mathbb{R}^n$ is the payoff vector in the subgame corresponding to the subgame strategy profile $\tilde{s} \in \tilde{S}$. We let $\hat{u}(\cdot; \tilde{s}) : \hat{S} \to \mathbb{R}^n$ be the payoff-vector function in the particular difference game $\hat{\Gamma}(\tilde{s})$ determined by a particular subgame strategy profile $\tilde{s} \in \tilde{S}$; i.e. $\hat{u}(\hat{s}; \tilde{s}) \in \mathbb{R}^n$ is the payoff vector in the difference game $\hat{\Gamma}(\tilde{s})$—defined by the subgame strategy profile $\tilde{s} \in \tilde{S}$—when the difference-game strategy profile $\hat{s} \in \hat{S}$ is played in the difference game.
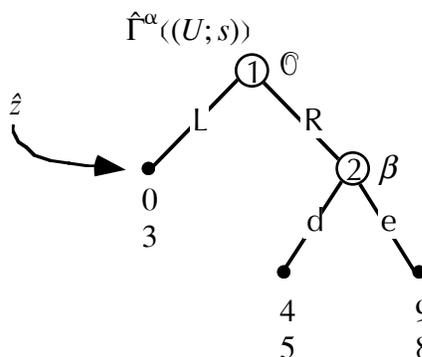
$$\hat{\Gamma}^\alpha((U;s))$$

Figure 5: The difference game determined by the particular choice
of subgame strategy profile $\tilde{s} = (U; s)$

Consider any subgame strategy profile $\tilde{s} \in \tilde{S}$ and the difference game $\hat{\Gamma}(\tilde{s})$ it defines. Now consider a difference-game strategy profile $\hat{s} \in \hat{S}$. This difference-game strategy profile results in a path in the difference game which either does or does not reach the node-of-decomposition x, and therefore either does or does not reach the subgame $\tilde{\Gamma}$. This dichotomy is useful in analyzing the difference-game payoff: If the subgame is reached by $\hat{s}$, the difference game's payoff is that of the subgame;[10] if the subgame is not reached by $\hat{s}$, the difference game's payoff is independent of the subgame strategy $\tilde{s}$. I.e.

$$\hat{u}(\hat{s}; \tilde{s}) \begin{cases} = \tilde{u}(\tilde{s}), & \hat{s} \text{ reaches } x, \\ \text{is independent of } \tilde{s}, & \hat{s} \text{ does not reach } x. \end{cases} \tag{2}$$

The payoff in the original game to the strategy profile $s = (\hat{s}, \tilde{s})$ is exactly the payoff to the difference-game strategy profile $\hat{s}$ in the difference game defined by the subgame strategy profile $\tilde{s}$: $\forall \tilde{s} \in \tilde{S}, \forall \hat{s} \in \hat{S}$, where $s = (\hat{s}, \tilde{s}) \in S$,

$$u(s) \equiv \hat{u}(\hat{s}; \tilde{s}). \tag{3}$$

To see this consider any strategy profile $s \in S$ of the original game Γ and consider the path of s. There are two cases: either 1 the path of s encounters the node-of-decomposition x or 2 it does not. 1 If the path of s encounters node x, then the path continues (as determined by the subgame strategy profile $\tilde{s}$) until it reaches a terminal node of the original game which is also a terminal node of the subgame $\tilde{\Gamma}$. In this case the payoff to s in the original game Γ is exactly the subgame payoff corresponding to the subgame strategy profile $\tilde{s}$; $u(s) = \tilde{u}(\tilde{s}) = \hat{u}(\hat{s}; \tilde{s})$, by (2). (The subgame payoffs corresponding to subgame terminal nodes are inherited from the original game's terminal-node payoffs.) 2 If the path of s does not reach the subgame, then the path of s is exactly the path of the difference-game strategy profile $\hat{s}$. (The path of s only encounters information sets in $\hat{H}$. Because $\hat{s}$ is the restriction of s to $\hat{H}$, s and $\hat{s}$ agree on $\hat{H}$.) Therefore, again thanks to the inheritance of payoffs from the original game, the payoff to s in the original game is exactly the payoff to $\hat{s}$ in the difference game; i.e. $u(s) = \hat{u}(\hat{s}; \tilde{s})$. Therefore we have

---

[10] If the subgame is reached, then the artificially added terminal node $\hat{z}$ is reached, and therefore the difference-game payoff is the payoff vector associated with $\hat{z}$. But this payoff is exactly the payoff of the subgame when the subgame strategy profile $\tilde{s}$ is played in the subgame.

established (3) in all cases.

Using (1) and (3) we see that

$$u((t_i, s_{-i})) = \hat{u}((\hat{t}_i, \hat{s}_{-i}); (\tilde{t}_i, \tilde{s}_{-i})). \tag{4}$$

Composing equilibria

The restriction of a Nash equilibrium to any on-the-equilibrium-path subgame is a Nash equilibrium of that subgame

Theorem 1   Consider an extensive-form game Γ, a subgame $\tilde{\Gamma}$ of Γ, and its associated difference extensive form $\hat{\Gamma}$. Let $s \in S$ be a Nash equilibrium of Γ and denote by $s = (\hat{s}; \tilde{s})$ the pair of difference-game and subgame strategy profiles which correspond to the equilibrium strategy profile s. If the difference-game strategy profile $\hat{s}$ reaches the subgame $\tilde{\Gamma}$, then the subgame strategy profile $\tilde{s}$ is a Nash equilibrium of the subgame $\tilde{\Gamma}$.

Proof   Because $\hat{s}$ reaches the subgame $\tilde{\Gamma}$, we know from (2) that the payoff to the entire strategy profile s is just the payoff to the subgame strategy profile $\tilde{s}$ in $\tilde{\Gamma}$; i.e. $\hat{u}(\hat{s}; \tilde{s}) = \tilde{u}(\tilde{s})$. Assume that $\tilde{s}$ is not a Nash equilibrium of $\tilde{\Gamma}$. Then there exists a player $i \in I$ and a subgame strategy $\tilde{t}_i \in \tilde{S}_i$ such that

$$\tilde{u}_i(\tilde{t}_i, \tilde{s}_{-i}) > \tilde{u}_i(\tilde{s}). \tag{5}$$

Let $\hat{t}_i = \hat{s}_i$ and let $t_i = (\hat{t}_i, \tilde{t}_i) \in S_i$. Using (4), $\hat{t}_i = \hat{s}_i$, (2), (5), (2), and (3), we write

$$u_i((t_i, s_{-i})) = \hat{u}_i((\hat{t}_i, \hat{s}_{-i}); (\tilde{t}_i, \tilde{s}_{-i})) = \hat{u}_i(\hat{s}; (\tilde{t}_i, \tilde{s}_{-i})) = \tilde{u}_i((\tilde{t}_i, \tilde{s}_{-i})) > \tilde{u}_i(\tilde{s}) = \hat{u}_i(\hat{s}; \tilde{s}) = u_i(s).$$

Therefore s, contrary to assertion, must not have been a Nash equilibrium of the original game Γ.   ☺

A subgame equilibrium and a difference-game equilibrium form an equilibrium of the entire game

Theorem 2   Consider an extensive-form game Γ, a subgame $\tilde{\Gamma}$ of Γ, and its associated difference extensive form $\hat{\Gamma}$. Let $\tilde{s}$ be a Nash equilibrium of the subgame $\tilde{\Gamma}$ and let $\hat{s}$ be a Nash equilibrium of the particular difference game $\hat{\Gamma}(\tilde{s})$. Then $s = (\hat{s}, \tilde{s})$ is a Nash equilibrium of Γ. (Kuhn [1953: 208].)

Proof   We need to show that for all players $i \in I$, and for all player-i strategies $t_i = (\hat{t}_i; \tilde{t}_i) \in S_i$,

$$u_i(s) \geq u_i(t_i, s_{-i}). \tag{6}$$

Consider the difference-game strategy profile $(\hat{t}_i, \hat{s}_{-i})$ implied by player i's proposed deviation. Because $\hat{s}$ is an equilibrium of $\hat{\Gamma}(\tilde{s})$, we have, using (3),

$$u_i(s) = \hat{u}_i(\hat{s}; \tilde{s}) \geq \hat{u}_i((\hat{t}_i, \hat{s}_{-i}); \tilde{s}). \tag{7}$$

There are two cases. 1 $(\hat{t}_i, \hat{s}_{-i})$ does not lead to the subgame $\tilde{\Gamma}$ and 2 $(\hat{t}_i, \hat{s}_{-i})$ does lead to the subgame $\tilde{\Gamma}$. In case 1, the difference-game payoff to $(\hat{t}_i, \hat{s}_{-i})$ is independent, using (2), of the subgame strategy profile played in $\tilde{\Gamma}$. Therefore, using (4), we have

$$\hat{u}_i((\hat{t}_i, \hat{s}_{-i}); \tilde{s}) = \hat{u}_i((\hat{t}_i, \hat{s}_{-i}); (\tilde{t}_i, \tilde{s}_{-i})) = u_i((t_i, s_{-i})). \tag{8}$$

Combining (7) and (8) yields (6), proving the proposition for case 1.

In case 2, the difference-game payoff is just the subgame payoff, using (2), so we have

$$\hat{u}_i((\hat{t}_i, \hat{s}_{-i}); \tilde{s}) = \tilde{u}_i(\tilde{s}). \tag{9}$$

Because $\tilde{s}$ is an equilibrium of $\tilde{\Gamma}$, using (2) and (4),

$$\tilde{u}_i(\tilde{s}) \geq \tilde{u}_i((\tilde{t}_i, \tilde{s}_{-i})) = \hat{u}_i((\hat{t}_i, \hat{s}_{-i}); (\tilde{t}_i, \tilde{s}_{-i})) = u_i((t_i, s_{-i})). \tag{10}$$

Combining (7), (9), and (10) yields the desired (6) for case 2.   ☺

Subgame perfection

We saw in the game of Figure 1 that Nash equilibria of an extensive-form game can fail to exhibit dynamic consistency (or sequential rationality): they can rely on specification of actions at off-the-equilibrium-path information sets which would not rationally be implemented if, contrary to specification, the off-the-path information set were reached.

Subgame perfection has been offered as a refinement of Nash equilibrium which in some cases successfully weeds out such undesirable equilibria. (Selten [1965], Selten [1975])

| Definition | A behavior strategy profile σ of an extensive-form game Γ is a *subgame-perfect* equilibrium of Γ if, for every subgame $\tilde{\Gamma}$ of Γ, the restriction of σ to the subgame $\tilde{\Gamma}$ is a Nash equilibrium of the subgame $\tilde{\Gamma}$.

We saw in Theorem 1 that the restriction of any Nash equilibrium to any on-the-equilibrium-path subgame is a Nash equilibrium of the subgame. Therefore the only additional restrictions being imposed by subgame perfection are at off-the-equilibrium-path subgames.

We see that subgame perfection is a refinement of Nash equilibrium because every subgame-perfect equilibrium is a Nash equilibrium. (Let σ be a subgame-perfect equilibrium of Γ. Every game Γ is a subgame of itself, so the restriction of σ to the subgame Γ, which is simply just σ, must be a Nash equilibrium of Γ.)

To see how subgame perfection is sometimes successful in eliminating dynamically inconsistent Nash

equilibria, let's revisit the game of Figure 1. That game had two pure-strategy Nash equilibria, viz. (D, l) and (U, r).[11] In Figure 6 I have indicated the only strictly smaller subgame of the whole game. We object to the (U, r) equilibrium because player 2 would not rationally choose r at his information set if he actually reached it. In order for (U, r) to be subgame perfect, its restriction to the subgame containing only player 2's information set—viz. the player-2 strategy r—must be a Nash equilibrium of that subgame. However, it fails that test.
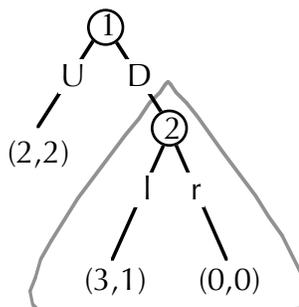

Figure 6: A subgame of the game from Figure 1.

Backward induction in games of perfect information

A finite game of perfect information has a pure-strategy equilibrium

We know that every finite game has a Nash equilibrium, however there need not exist a pure-strategy equilibrium. When the game has perfect information, however, the existence of a pure-strategy equilibrium is assured. In a game of perfect information every decision node $x \in X$ is a node at which the game can be decomposed into a subgame.

| **Theorem 3** | Every finite game of perfect information has a pure-strategy equilibrium. (Kuhn [1953: 209].) |

Proof   We will prove this by induction on the number m of decision nodes in Γ; i.e. let $m = \#X$. If there is only one decision node, i.e. $m = 1$, then there exists a pure-strategy equilibrium in which the player who owns that node chooses an action which maximizes her payoff. (The other players, if any, have empty strategy spaces.) Now we establish that if every perfect-information game with less than m decision nodes has a pure-strategy equilibrium, then every game with exactly m decision nodes has a pure-strategy equilibrium. Then by induction, starting with the existence of a pure-strategy equilibrium for $m = 1$, all finite games of perfect information have pure-strategy equilibria.

Let $m > 1$. Because Γ has perfect information, we can decompose it at some noninitial decision node

---

[11]   There is also a continuum of equilibria in which player 1 chooses U and player 2 chooses l with any probability weakly less than @/3. The intuition behind this continuum is the following: As long as player 1 chooses U, player 2 is indifferent to whether he chooses l or r. As long as player 2 does not choose l with too high a probability, U is a best response by player 1. Only when player 2 chooses l more than @/3 of the time will player 1's expected payoff to D exceed the 2 she receives for certain by playing U.

$x \in X \setminus \{\mathbb{0}\}$ into a subgame $\tilde{\Gamma}$ with $m' < m$ decision nodes and a difference extensive form $\hat{\Gamma}$ with $m - m' < m$ decision nodes. (Both the subgame and the difference extensive form have perfect information.) Because every perfect-information game with less than m decision nodes has a pure-strategy equilibrium, $\tilde{\Gamma}$ does. Let $\tilde{s}$ be a pure-strategy equilibrium of $\tilde{\Gamma}$. Now consider the difference game $\hat{\Gamma}(\tilde{s})$ determined by this equilibrium subgame strategy profile. It is a perfect-information game with fewer than m decision nodes. Therefore it has a pure-strategy equilibrium. Let $\hat{s}$ be a pure-strategy equilibrium of $\hat{\Gamma}(\tilde{s})$. Then, by Theorem 2, $s = (\hat{s}, \tilde{s})$ is an equilibrium of the original game Γ. Therefore every perfect-information game with m decision nodes has a pure-strategy equilibrium. ☺

Zermelo's backward-induction algorithm

In order to solve a perfect-information game by backward induction we first need to identify the set of *penultimate* decision nodes. A decision node is penultimate if all of its immediate successors are terminal nodes. A subgame which has a penultimate decision node x as its initial node is just a single-player game; the decision problem belongs to the owner $\iota(x)$ of that node. That player will choose some action available at that node which maximizes her payoff over the terminal nodes which can reached from x. After we specify the player's choice at x, we can delete the successors of x from the game. Then we change x from a decision node to a terminal node, assigning the new terminal node x the payoff vector associated with the former terminal node reached by the stipulated action.[12]

This process shrinks the game. We identify the new set of penultimate decision nodes in this smaller game and repeat the above process, keeping track of each player's chosen action at each of her penultimate decision nodes. Eventually the game is reduced until there is only one decision node left, viz. the initial node $\mathbb{0}$. The owner of the initial node then makes her payoff-maximizing decision and this completes the determination of a backward-induction solution of the game. By this time every decision node in the original game has become a penultimate decision node for its owner, and so an action has been recorded for it. The union of all these decisions constitutes the strategy profile corresponding to the backward-induction solution. (This algorithm is described as a flowchart in Figure 7.)

For any decision node $x \in X$, let $N(x) \subset V$ be the set of immediate successors of x. Let $\check{v}(a \mid x) \in N(x)$ be the node reached directly when action $a \in A(x)$ is taken at node x.

Let $\overline{X} \subset X$ and $\overline{Z} \subset \overline{V}$ be sets of decision nodes and terminal nodes, respectively. Then the set $P(\overline{X}, \overline{Z})$ of penultimate decision nodes are those decision nodes whose immediate successors are all terminal nodes:

$$P(\overline{X}, \overline{Z}) = \{x \in \overline{X} : N(x) \subset \overline{Z}\}.$$

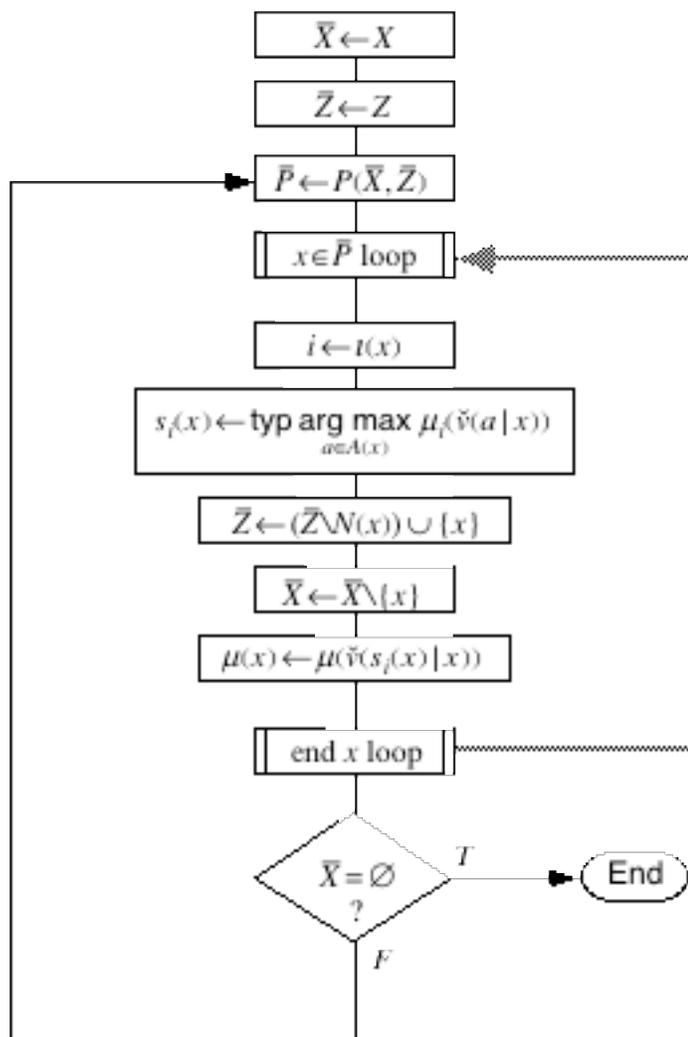---

[12]   This algorithm has been attributed to Zermelo [1912].

Figure 7: Zermelo's backward-induction algorithm.

Recall that $\mu: Z \to \mathbb{R}^n$ specifies the assignment of payoff vectors to terminal nodes in the original game. The backward-induction process will extend the domain of definition of μ so that it will also assign payoff vectors to each decision node at the time it makes the transition from penultimate decision node to new terminal node.

We let the sets of decision and terminal nodes at any point in the process be $\overline{X}$ and $\overline{Z}$, respectively. We initialize the process by assigning these to be the corresponding sets in the original game, i.e. $\overline{X} = X$ and $\overline{Z} = Z$. Let $\overline{P}$ be the set of penultimate decision nodes at any stage of the backward-induction process. A penultimate decision node $x \in \overline{P}$ belongs to player $i = \iota(x)$. Player i will choose an action $a \in A(x)$ available at this node such that

$$a \in \arg\max_{a' \in A(x)} \mu_i(\check{v}(a' \,|\, x)).$$

We record this choice made at node x by setting $s_i(x) = a$.[13]

Now we remove all of x's immediate successors $N(x)$ from the game's set of terminal nodes and replace them in the set of terminal nodes with x; i.e. $\bar{Z} \leftarrow (Z \backslash N(x)) \cup \{x\}$. Now that x is no longer a decision node, we remove x from the set of decision nodes; i.e. $\bar{X} \leftarrow \bar{X} \backslash \{x\}$. The payoff vector we assign to the new terminal node x is the payoff vector associated with the terminal node $\check{v}(s_i(x)|x)$ reached when player i takes action $s_i(x)$ at node x; i.e. $\mu(x) = \mu(\check{v}(s_i(x)|x))$.

Every decision node eventually becomes a penultimate decision for its owner, an action is recorded for it, and it becomes a terminal node. The last decision node to become an penultimate decision node is the initial node $\mathbb{O}$. When an action is determined for the initial node it is removed from the set $\bar{X}$ of current decision nodes, making this set empty. This is the stopping test for the algorithm.

Theorem 2 can be used to show that any strategy profile derived by Zermelo's algorithm in a game of perfect information is a subgame-perfect equilibrium of that game.

References

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

Selten, Reinhard [1965] "Spieltheoretische Behandlung eines Oligopolmodells mit Nachfrageträgheit," *Zeitschrift für die gesamte Staatswissenschaft* **121**: 301–324, 667–689.

Selten, Reinhard [1975] "Reexamination of the Perfectness Concept for Equilibrium Points in Extensive Games," *International Journal of Game Theory* **4** 1: 25–55.

Zermelo, E. [1912] "Über eine Anwendung der Mengenlehre auf die Theorie des Schachspiels," *Proceedings of the Fifth International Congress on Mathematics, Cambridge* **Vol. 2**: 501.

---

[13]   The function name typ in the flowchart might be interpreted as "typical" or "t[ake] y[our] p[ick]." It is, of course, crucial that typ truly be a function, i.e. uniquely defined. The only other important property of typ S, for $S \neq \varnothing$, is that typ $S \in S$. In this context it is just a way of expressing that we need a definite member of the arg max, but we don't care which member it is.