Perfect-Information, Extensive-Form Games

Extensive Form Game

- Informally speaking, a tree, where each node represents the choice of one of the players. So, turn-based game, with a concept of order of actions
- Leaves represent final outcomes over which each player has a utility function

Definition 5.1.1 (Perfect-information game) *A (finite) perfect-information game (in extensive form) is a tuple* $G = (N, A, H, Z, \chi, \rho, \sigma, u)$*, where:*

- N *is a set of* n *players;*
- A *is a (single) set of actions;*
- H *is a set of nonterminal choice nodes;*
- Z *is a set of terminal nodes, disjoint from* H*;*
- $\chi : H \mapsto 2^A$ *is the action function, which assigns to each choice node a set of possible actions;*
- $\rho : H \mapsto N$ *is the player function, which assigns to each nonterminal node a player* $i \in N$ *who chooses an action at that node;*
- $\sigma : H \times A \mapsto H \cup Z$ *is the successor function, which maps a choice node and an action to a new choice node or terminal node such that for all* $h_1, h_2 \in H$ *and* $a_1, a_2 \in A$*, if* $\sigma(h_1, a_1) = \sigma(h_2, a_2)$ *then* $h_1 = h_2$ *and* $a_1 = a_2$*; and*
- $u = (u_1, \ldots, u_n)$*, where* $u_i : Z \mapsto \mathbb{R}$ *is a real-valued utility function for player* i *on the terminal nodes* Z*.*

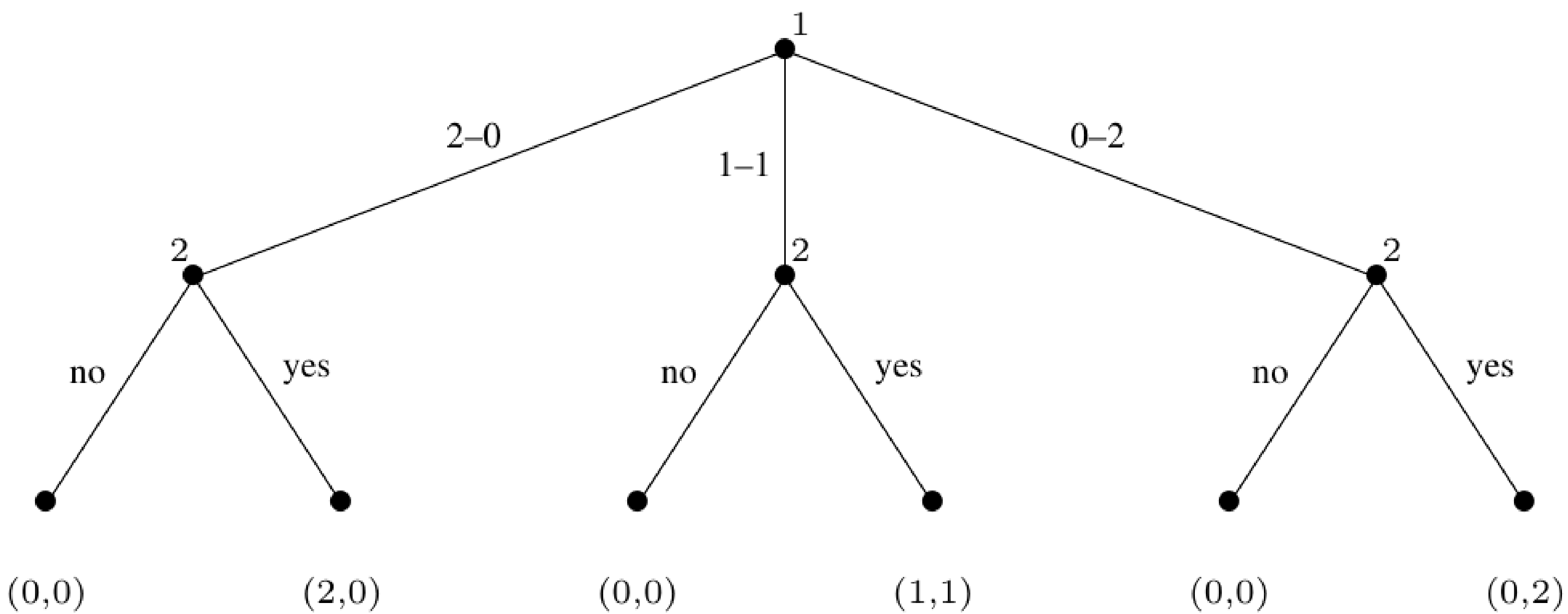

Figure 5.1: The Sharing game.

Strategies and Equilibria in Extensive-Form Games

Strategies

- A pure strategy is complete specification of choice of made of each player at every node

Definition 5.1.2 (Pure strategies) *Let* $G = (N, A, H, Z, \chi, \rho, \sigma, u)$ *be a perfect-information extensive-form game. Then the pure strategies of player* i *consist of the Cartesian product* $\prod_{h \in H, \rho(h)=i} \chi(h)$.

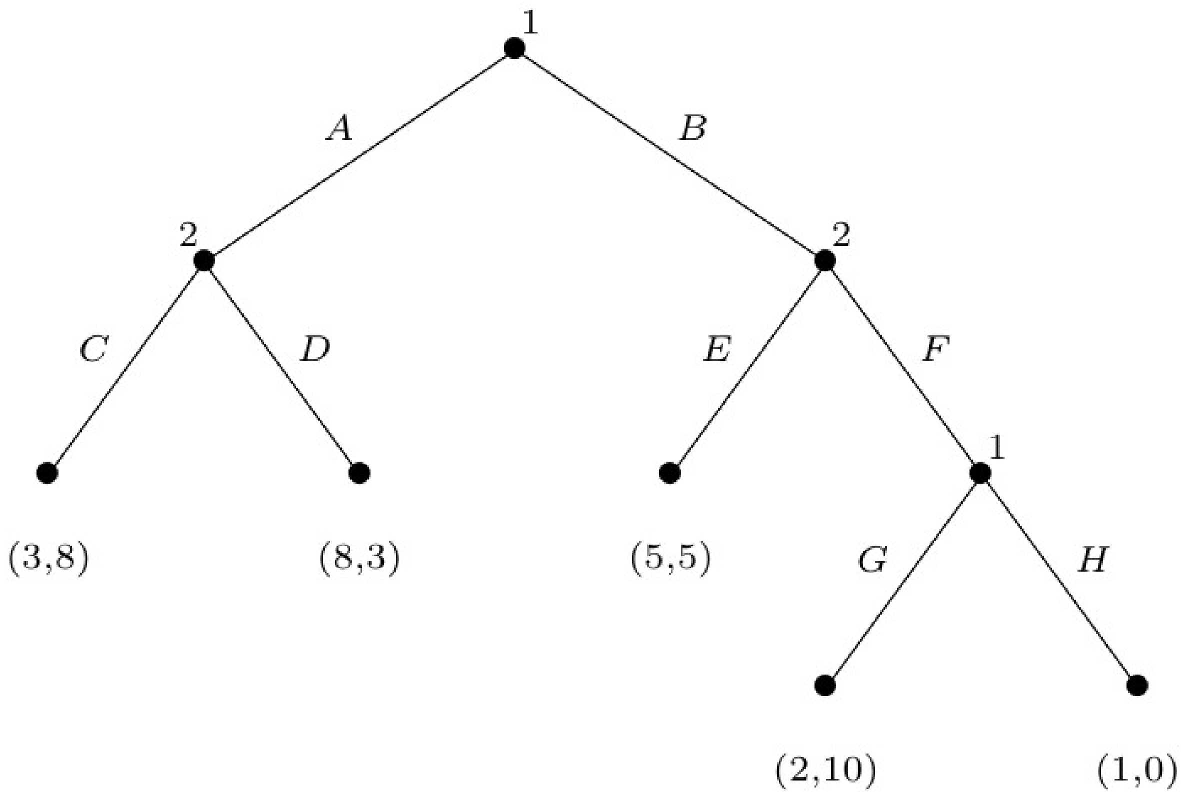

Figure 5.2: A perfect-information game in extensive form.

$$S_1 = \{(A, G), (A, H), (B, G), (B, H)\}$$

$$S_2 = \{(C, E), (C, F), (D, E), (D, F)\}$$

	(C,E)	(C,F)	(D,E)	(D,F)
(A,G)	3, 8	3, 8	8, 3	8, 3
(A,H)	3, 8	3, 8	8, 3	8, 3
(B,G)	5, 5	2, 10	5, 5	2, 10
(B,H)	5, 5	1, 0	5, 5	1, 0

Figure 5.3: The game from Figure 5.2 in normal form.

- Every perfect-information, EF game has normal-form representation. But note redundancy

- However, not every normal-form game has a extensive-form representation
- Consider Prisoner's Dilemma

	C	D
C	$-1, -1$	$-4, 0$
D	$0, -4$	$-3, -3$

Figure 3.3: The TCP user's (aka the Prisoner's) Dilemma.

Theorem 5.1.3 *Every (finite) perfect-information game in extensive form has a pure-strategy Nash equilibrium.*

Subgame-Perfect Equilibria

Example Game

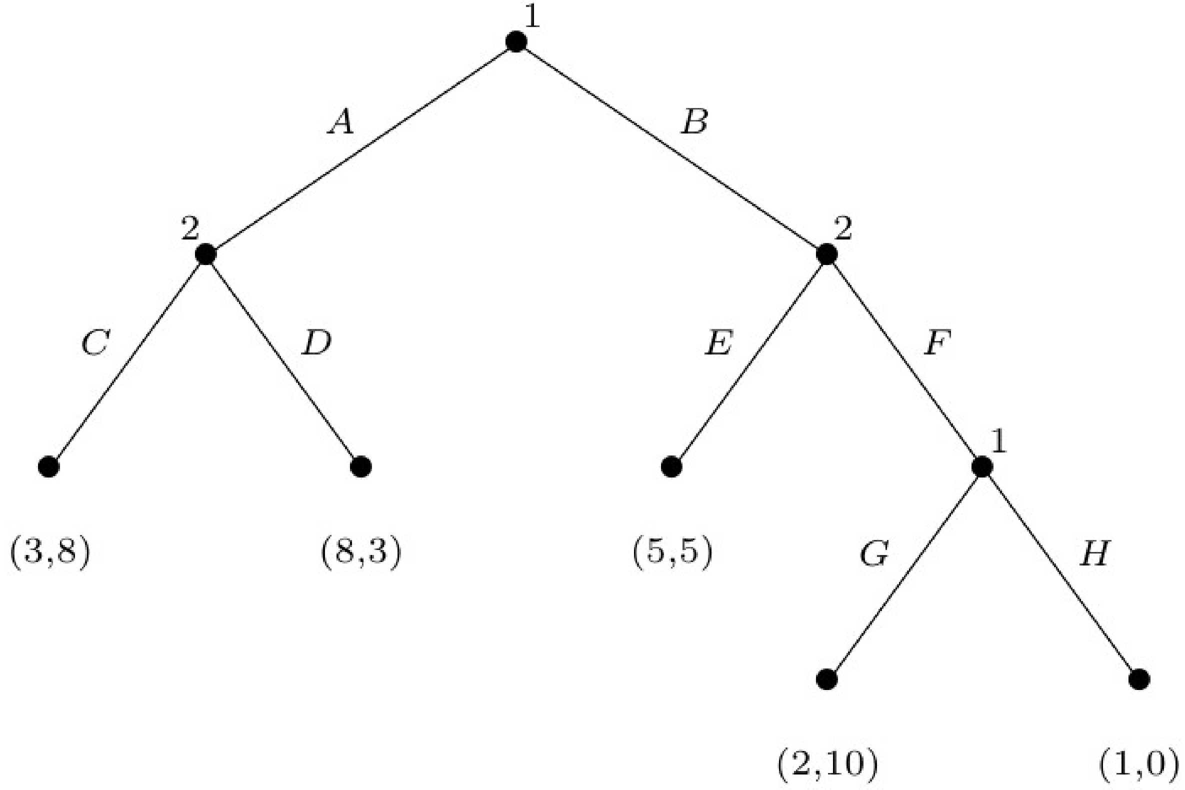

Figure 5.2: A perfect-information game in extensive form.

Pure-Strategy Nash Equilibria

	(C, E)	(C, F)	(D, E)	(D, F)
(A, G)	3, 8	$\boxed{3, 8}$	8, 3	8, 3
(A, H)	3, 8	$\boxed{3, 8}$	8, 3	8, 3
(B, G)	5, 5	2, 10	5, 5	2, 10
(B, H)	$\boxed{5, 5}$	1, 0	5, 5	1, 0

Figure 5.4: Equilibria of the game from Figure 5.2.

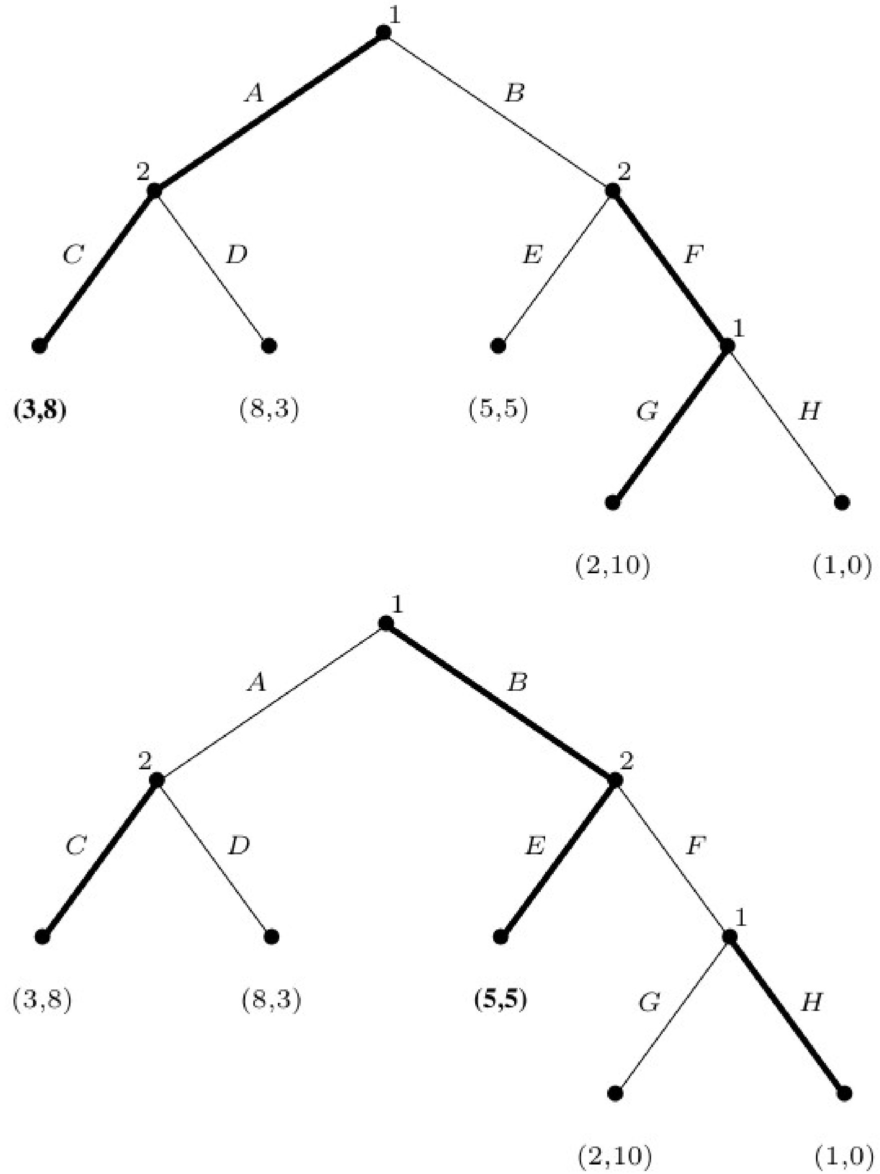

Figure 5.5: Two out of the three equilibria of the game from Figure 5.2: $\{(A, G), (C, F)\}$ and $\{(B, H), (C, E)\}$. Bold edges indicate players' choices at each node.

Pure-Strategy Nash Equilibria

Definition 5.1.4 (Subgame) *Given a perfect-information extensive-form game G, the* subgame *of G rooted at node h is the restriction of G to the descendants of h. The set of subgames of G consists of all of subgames of G rooted at some node in G.*

Definition 5.1.5 (Subgame-perfect equilibrium) *The* subgame-perfect equilibria *(SPE) of a game G are all strategy profiles s such that for any subgame G' of G, the restriction of s to G' is a Nash equilibrium of G'.*

Computing equilibria: backward induction

How to compute SPE?

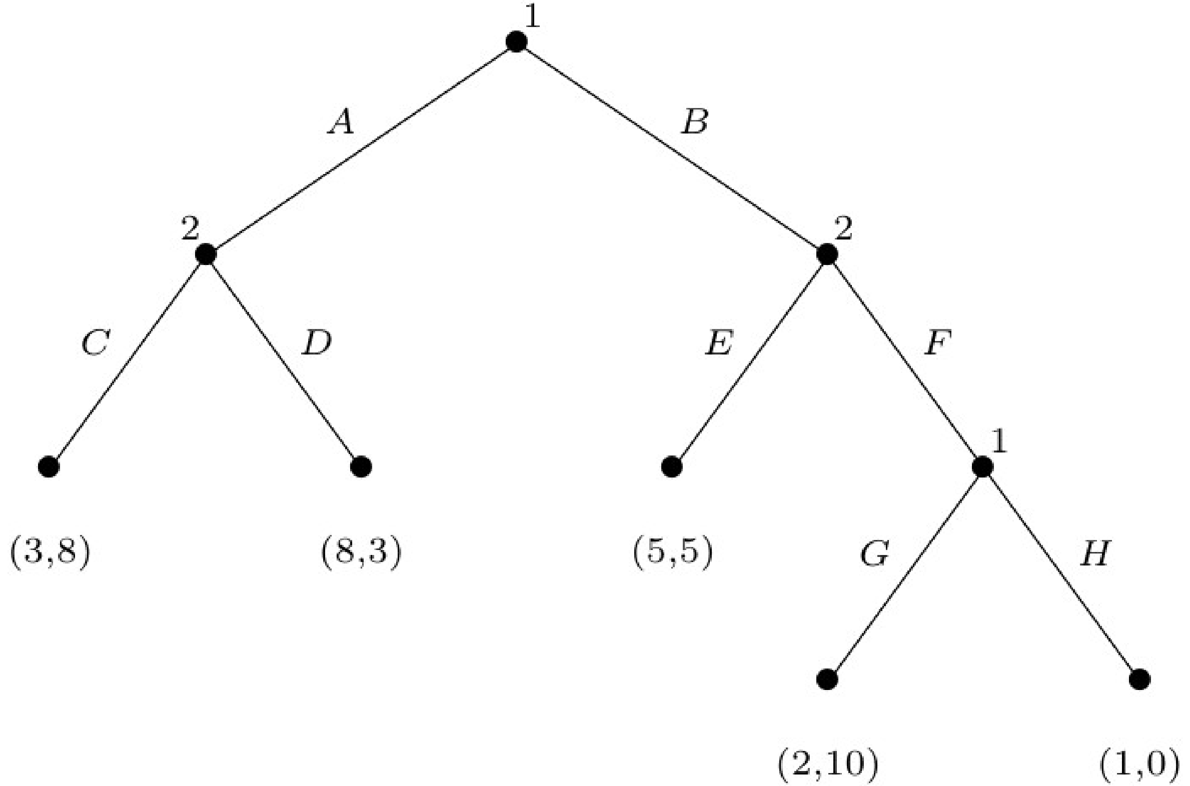

Figure 5.2: A perfect-information game in extensive form.

Backward Induction

function BACKWARDINDUCTION (node h) **returns** $u(h)$
if $h \in Z$ **then**
  return $u(h)$ // h is a terminal node
$best\_util \leftarrow -\infty$
forall $a \in \chi(h)$ **do**
  $util_at_child \leftarrow$BACKWARDINDUCTION$(\sigma(h, a))$
  if $util\_at\_child_{\rho(h)} > best\_util_{\rho(h)}$ **then**
    $best\_util \leftarrow util\_at\_child$
return $best\_util$

Figure 5.6: Procedure for finding the value of a sample (subgame-perfect) Nash equilibrium of a perfect-information extensive-form game.

Backward Induction

- In principle, a sample SPE is effectively computable
- In practice, game tree not enumerated in advance
- Extensive form representation of chess has around 10^{150} nodes

Alpha-Beta Pruning

- In 2-player, zero-sum game, we can prune away subtrees without examining the entire subtree

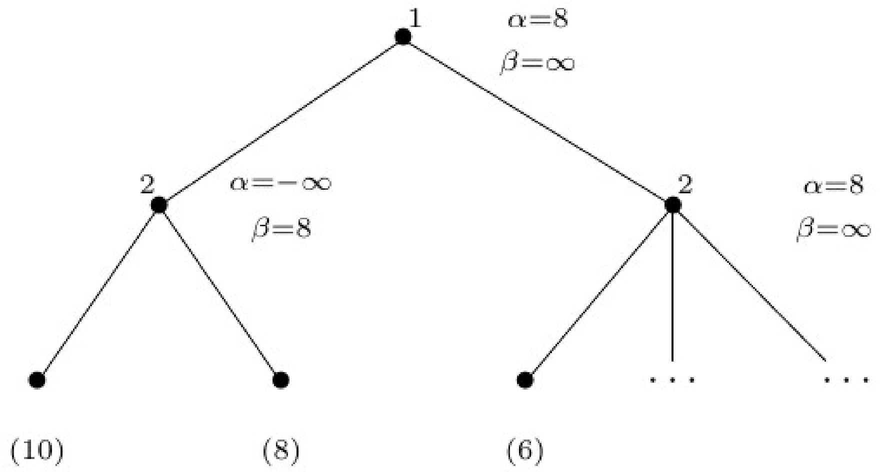

Figure 5.8: An example of alpha-beta pruning. We can backtrack after expanding the first child of the right choice node for player 2.

Alpha-Beta Pruning

- In 2-player, zero-sum game, we can prune away subtrees without examining the entire subtree
- Best case: $O(b^{m/2})$ time complexity. Random case: $O(b^{3m/4})$
- Exponential improvement, but still infeasible for something like chess
- In practice, engines do a limited depth alpha-beta pruning, using some evaluation function for an internal node as if it were a leaf

Backward Induction, Criticism

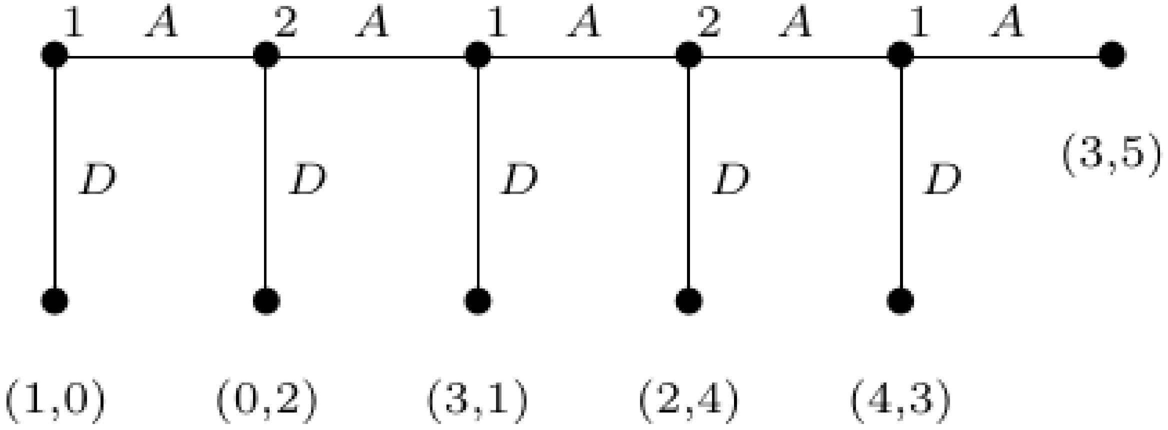

Figure 5.9: The Centipede game.