# CAP 5993/CAP 4993
# Game Theory

**Instructor: Sam Ganzfried**

**sganzfri@cis.fiu.edu**

# Golden Balls: the weirdest split or steal ever!

- https://www.youtube.com/watch?v=S0qjK3TWZE8

# Announcements

- HW1
- HW2 due 2/21, added in problems from lecture

# Game trees

- A tree is a triple G = (V, E, $x^0$) where (V,E) is a directed graph, $x^0$ in V is a vertex called the *root* of the tree, and for every vertex x in V there is a unique path in the graph from $x^0$ to x.
- Various games can be represented as trees. When a tree represents a game, the root of the tree corresponds to the *initial position* of the game, and every *game position* is represented by a vertex of the tree. The children of each vertex v are the vertices corresponding to the game positions that can be arrived at from v via one action. In other words, the number of children of a vertex is equal to the number of possible actions in the game position corresponding to that vertex.
  - For every vertex that is not a leaf, we need to specify the player who is to take an action at that vertex
  - At each leaf, we need to describe the outcome of the game.

- A game in extensive form (or extensive-form game) is an ordered vector $\Gamma = (N, V, E, x^0, (V_i)$ i in N, O, u)
  - N is finite set of players
  - $(V, E, x^0)$ is a tree called the *game tree*
  - $(V_i)$ i in N is a partition of the set of vertices that are not leaves.
  - O is the set of possible game outcomes.
  - u is a utility function associating every leaf of the tree with a game outcome in the set O.
- Let B be a nonempty set. A *partition* of B is a collection $B_1, B_2, \ldots, B_K$ of pairwise disjoint and nonempty subsets of B whose union is B.

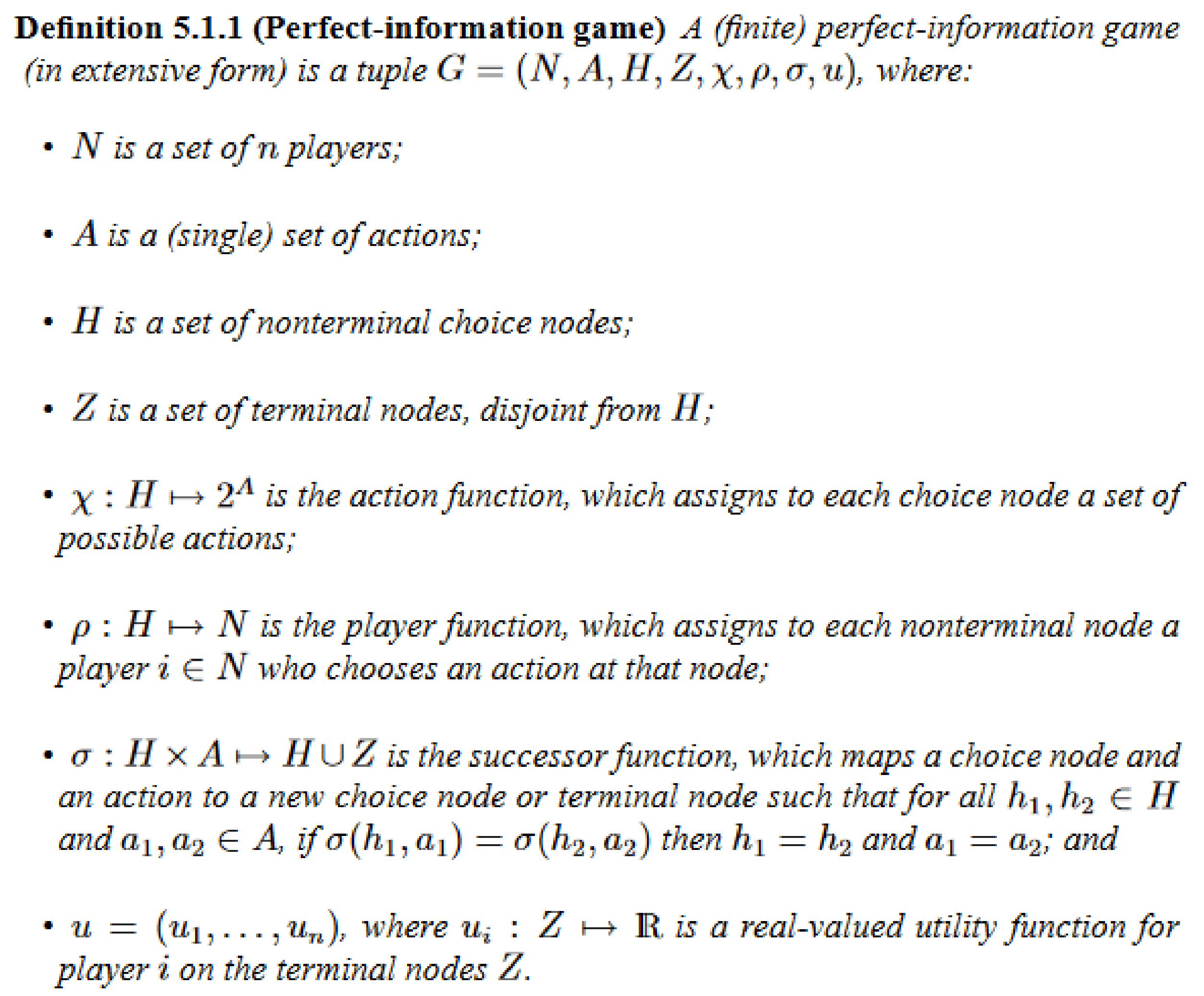
- It follows from the above description that every player who is to take an action knows the current state of the game, meaning that he knows all the actions in the game that led to the current point in the play. This implicit assumption is called *perfect information.*

- A strategy for a player i is a function $s_i$ mapping each vertex x in $V_i$ to an element in A(x) (equivalently, to an element in C(x)).
- According to this definition, a strategy includes instructions on how to behave at each vertex in the game tree, including vertices that previous actions by the player preclude from being reached. For example, in the game of chess, even if White's strategy calls for opening by moving a pawn from c2 to c3, the strategy must include instructions on how White should play his second move if in his first move he instead moved a pawn from c2 to c4, and Black then took his action.

- A strategy vector is a list of strategies s = ($s_i$) i in N, one for each player. Player i's set of strategies is denoted by $S_i$, and the set of all strategy vectors is denoted S = $S_1$x $S_2$ x…x $S_n$. Every strategy vector determines a unique play from the root to a leaf.
- Let Γ = (N, V, E, $x^0$, ($V_i$) i in N, O, u) be an extensive-form game (with perfect information), and let x in V be a vertex in the game tree. The *subgame starting at x*, denoted by Γ(x), is the extensive-form game Γ(x) = (N, V(x), E(x), x, ($V_i$(x)) i in N, O, u).
  - V(x) includes x and all vertices that are descendants of x.

- Theorem (von Neumann [1928]) In every two-player game (with perfect information) in which the set of outcomes is O = {Player 1 wins, Player 2 wins, Draw}, one and only one of the following three alternatives holds:
  1. Player 1 has a winning strategy.
  2. Player 2 has a winning strategy.
  3. Each of the two players has a strategy guaranteeing at least a draw.

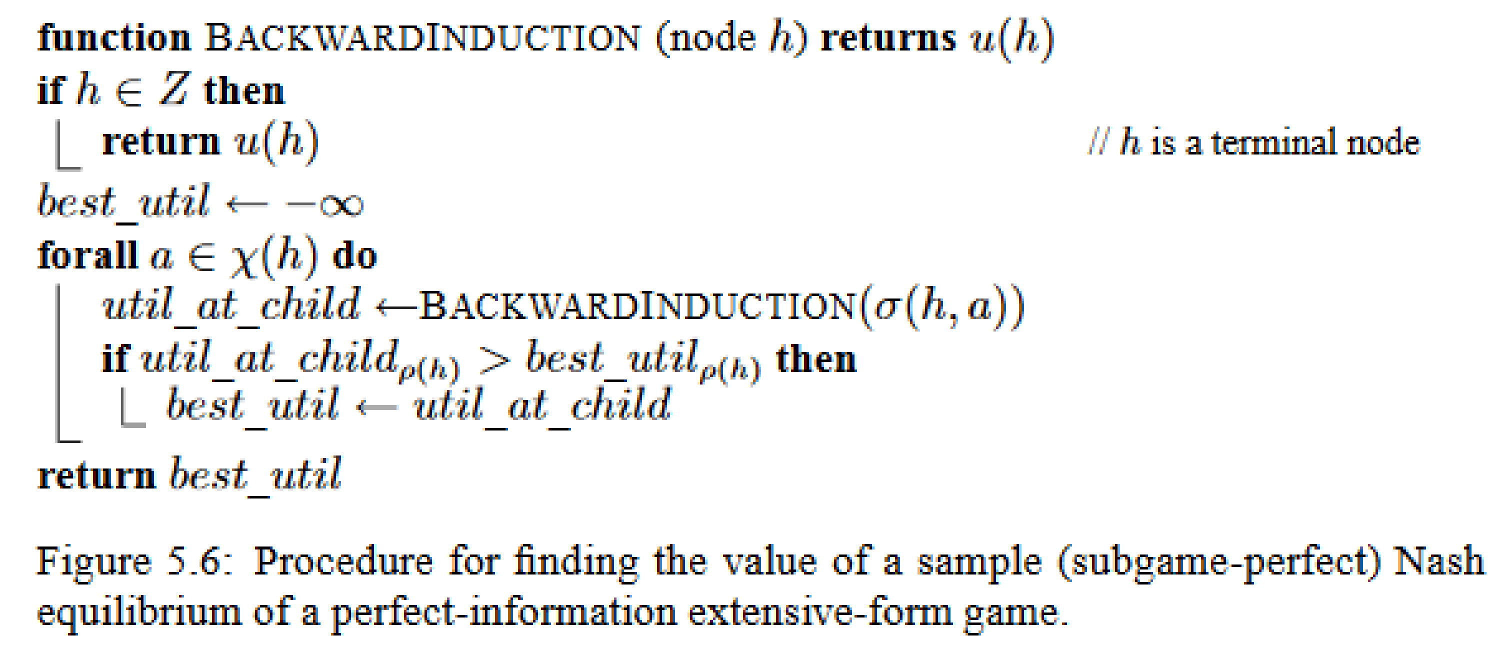
- Theorem (Kuhn) Every finite game with perfect information has at least one pure strategy Nash equilibrium.
  - “This is perhaps the earliest result in game theory, due to Zermelo in 1913).
- Corollary of Nash’s Theorem: Every extensive-form game (of perfect or imperfect information) has an equilibrium in mixed strategies.



**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* $Z$*.*

# 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.

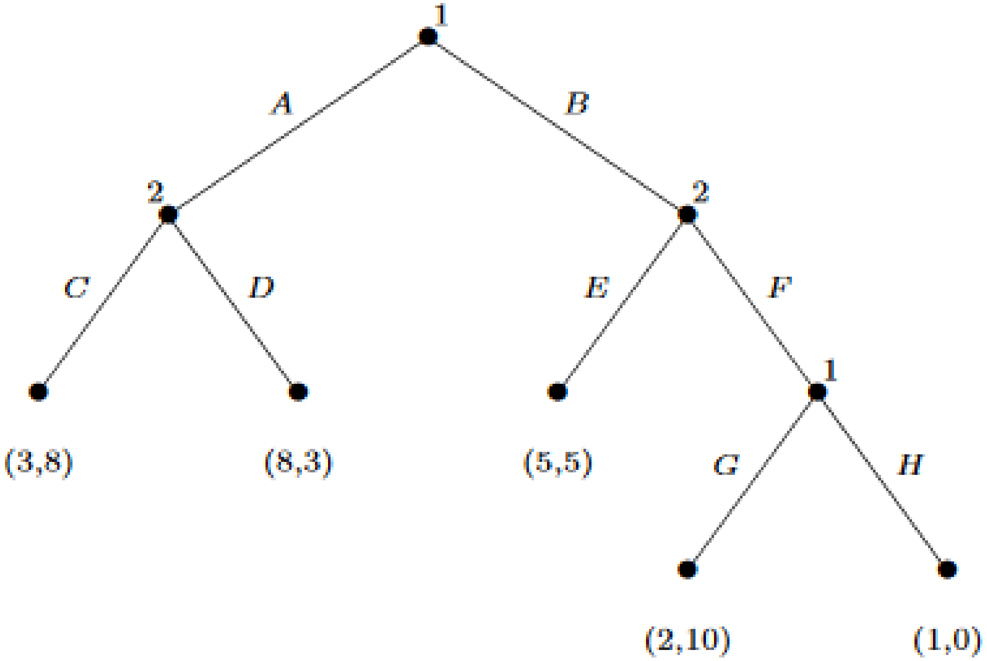
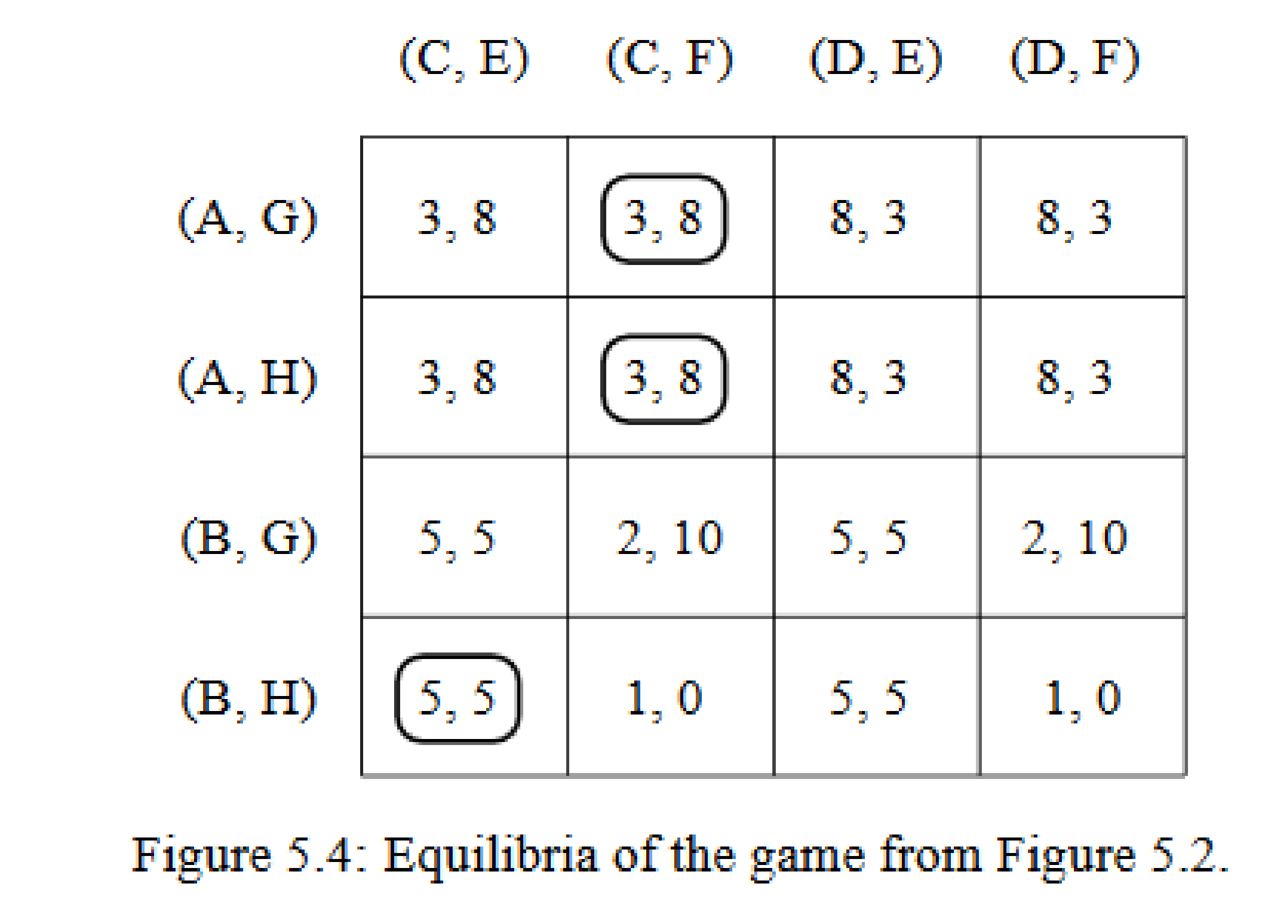


Figure 5.2: A perfect-information game in extensive form.



|  | (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.

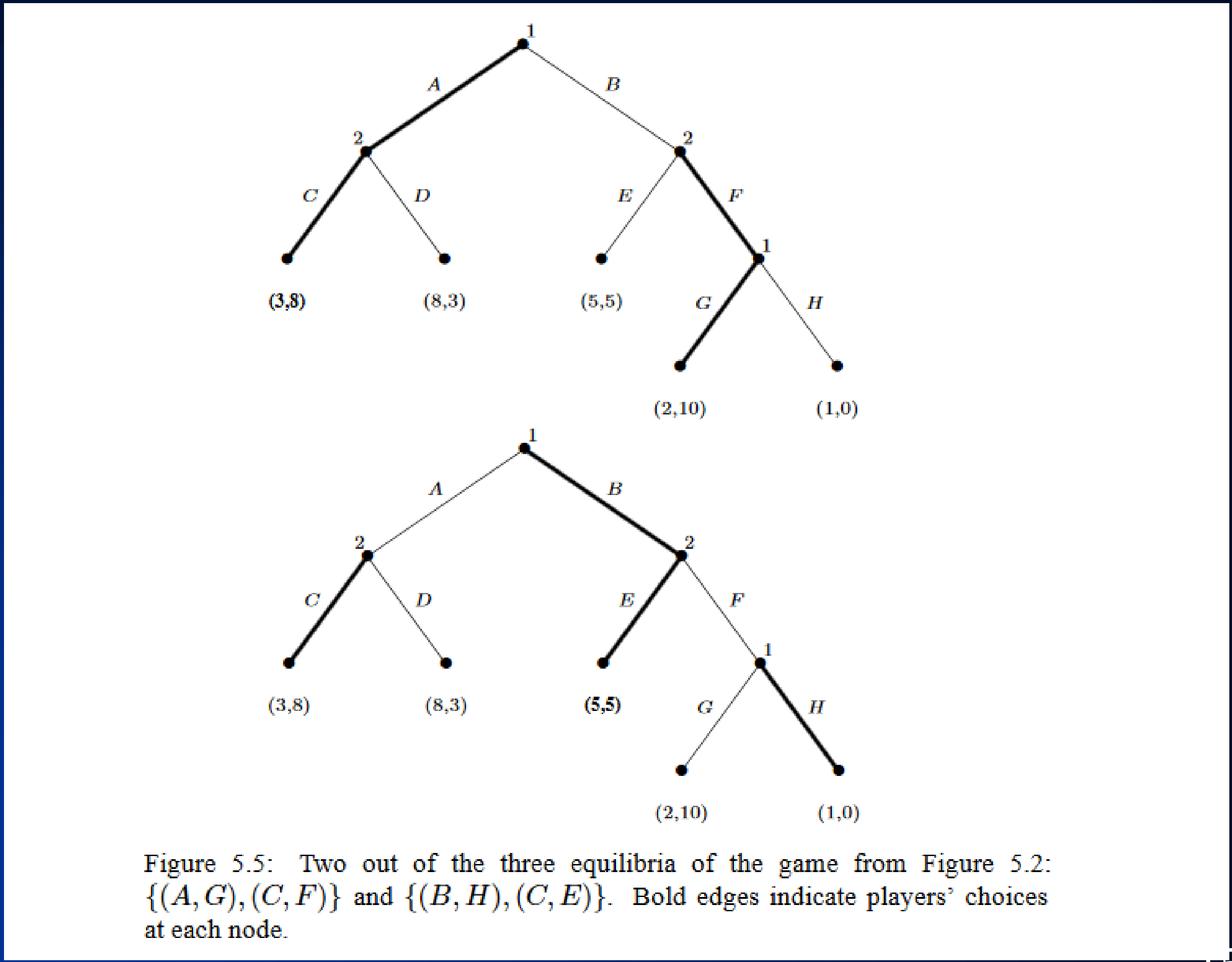


Figure 5.5: Two out of the three equilibria of the game from Figure 5.2: $\{(A,G),(C,F)\}$ and $\{(B,H),(C,E)\}$. Bold edges indicate players' choices at each node.

- First consider {(A,G),(C,F)}.
- {(B,H), (C,E)} less intuitive.
- Why is {(B,G), (C,E)} not an equilibrium??
- Player 1's decision to play H is a **threat**. But is it **credible**?

- Definition: 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 subgames of G rooted at some node in G.

- The **subgame-perfect equilibrium (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'.

# Subgame perfect equilibrium

- Every SPE is also a Nash equilibrium
- Furthermore, although SPE is a stronger concept than Nash equilibrium (i.e., every SPE is a Nash equilibrium, but not every NE is a SPE) it is still the case that every perfect-information extensive-form game has at least one subgame-perfect equilibrium.
- Rules out "noncredible threats." The only SPE is {(A,G, (C,F)}. Consider the subgame rooted at player 1's second choice node …

# Two-player zero-sum extensive-form games with imperfect information

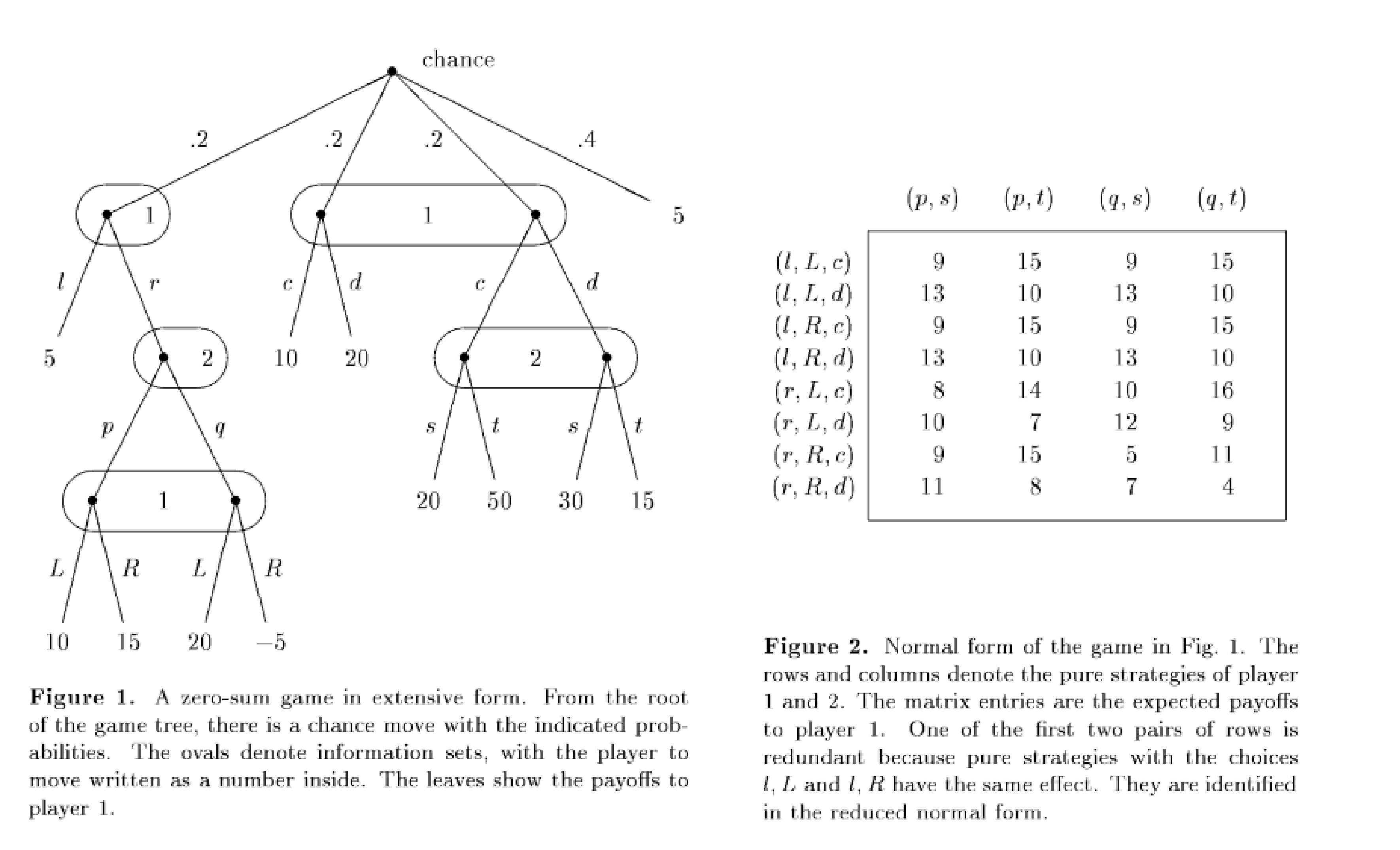
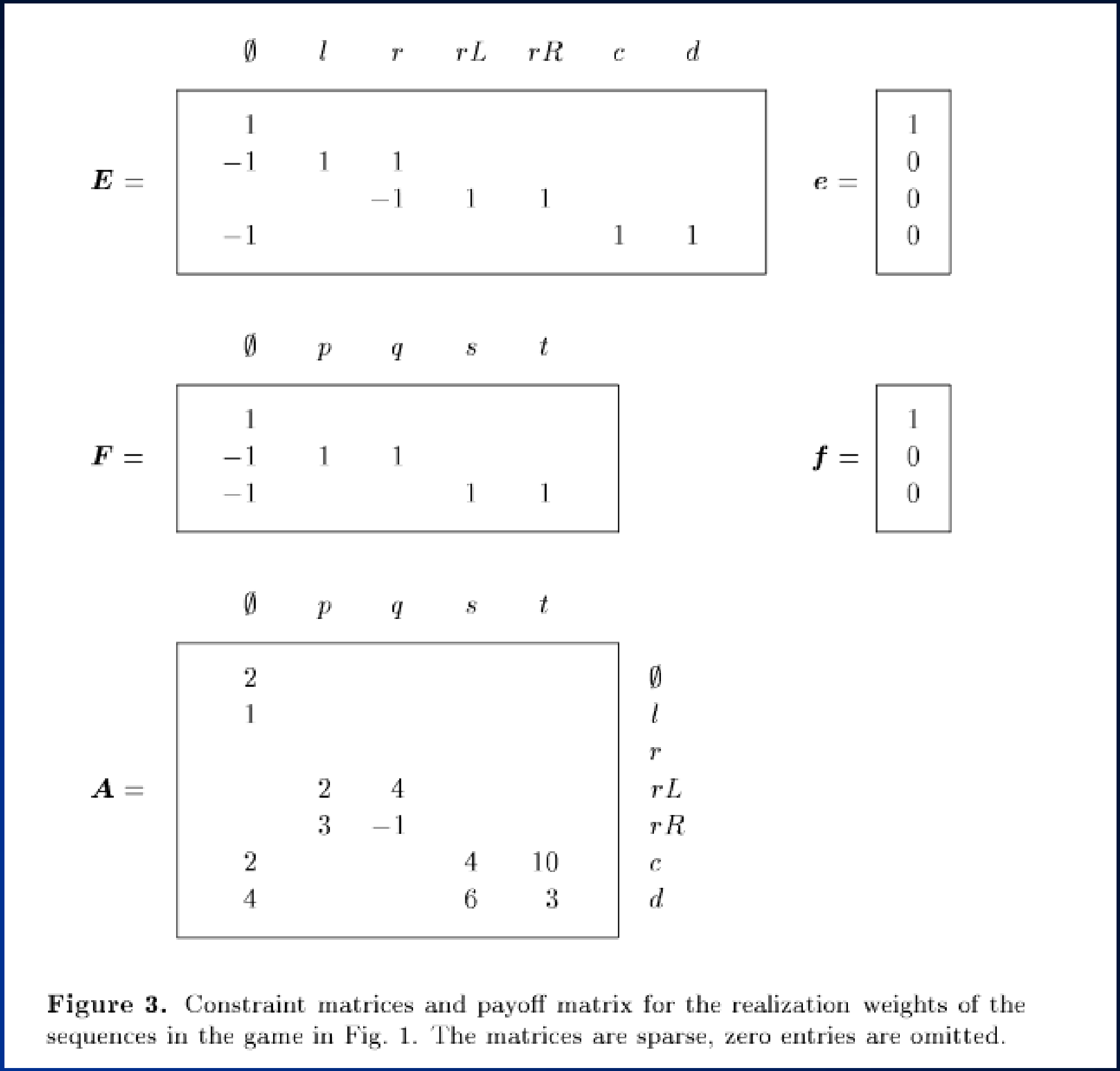


**Figure 1.** A zero-sum game in extensive form. From the root of the game tree, there is a chance move with the indicated probabilities. The ovals denote information sets, with the player to move written as a number inside. The leaves show the payoffs to player 1.

| | $(p,s)$ | $(p,t)$ | $(q,s)$ | $(q,t)$ |
|---|---|---|---|---|
| $(l,L,c)$ | 9 | 15 | 9 | 15 |
| $(l,L,d)$ | 13 | 10 | 13 | 10 |
| $(l,R,c)$ | 9 | 15 | 9 | 15 |
| $(l,R,d)$ | 13 | 10 | 13 | 10 |
| $(r,L,c)$ | 8 | 14 | 10 | 16 |
| $(r,L,d)$ | 10 | 7 | 12 | 9 |
| $(r,R,c)$ | 9 | 15 | 5 | 11 |
| $(r,R,d)$ | 11 | 8 | 7 | 4 |

**Figure 2.** Normal form of the game in Fig. 1. The rows and columns denote the pure strategies of player 1 and 2. The matrix entries are the expected payoffs to player 1. One of the first two pairs of rows is redundant because pure strategies with the choices $l, L$ and $l, R$ have the same effect. They are identified in the reduced normal form.



$$
\boldsymbol{E} = \begin{array}{ccccccc}
\emptyset & l & r & rL & rR & c & d \\
\hline
1 & & & & & & \\
-1 & 1 & 1 & & & & \\
 & & -1 & 1 & 1 & & \\
-1 & & & & & 1 & 1
\end{array}
\qquad
\boldsymbol{e} = \begin{array}{c} 1 \\ 0 \\ 0 \\ 0 \end{array}
$$

$$
\boldsymbol{F} = \begin{array}{ccccc}
\emptyset & p & q & s & t \\
\hline
1 & & & & \\
-1 & 1 & 1 & & \\
-1 & & & 1 & 1
\end{array}
\qquad
\boldsymbol{f} = \begin{array}{c} 1 \\ 0 \\ 0 \end{array}
$$

$$
\boldsymbol{A} = \begin{array}{ccccc|c}
\emptyset & p & q & s & t & \\
\hline
2 & & & & & \emptyset \\
1 & & & & & l \\
 & & & & & r \\
 & 2 & 4 & & & rL \\
 & 3 & -1 & & & rR \\
2 & & & 4 & 10 & c \\
4 & & & 6 & 3 & d
\end{array}
$$

**Figure 3.** Constraint matrices and payoff matrix for the realization weights of the sequences in the game in Fig. 1. The matrices are sparse, zero entries are omitted.

# LP formulation

$$\begin{array}{lrcl}
\underset{\boldsymbol{y},\,\boldsymbol{p}}{\text{minimize}} & \boldsymbol{e}^T\boldsymbol{p} & & \\
\text{subject to} & -\boldsymbol{A}\boldsymbol{y} + \boldsymbol{E}^T\boldsymbol{p} & \geq & 0 \\
 & -\boldsymbol{F}\boldsymbol{y} & = & -\boldsymbol{f}\,, \\
 & \boldsymbol{y} & \geq & 0\,.
\end{array} \qquad (8)$$

$$\begin{array}{lrcl}
\underset{\boldsymbol{x},\,\boldsymbol{q}}{\text{maximize}} & -\boldsymbol{q}^T\boldsymbol{f} & & \\
\text{subject to} & \boldsymbol{x}^T(-\boldsymbol{A}) - \boldsymbol{q}^T\boldsymbol{F} & \leq & 0 \\
 & \boldsymbol{x}^T\boldsymbol{E}^T & = & \boldsymbol{e}^T, \\
 & \boldsymbol{x} & \geq & 0\,.
\end{array} \qquad (9)$$

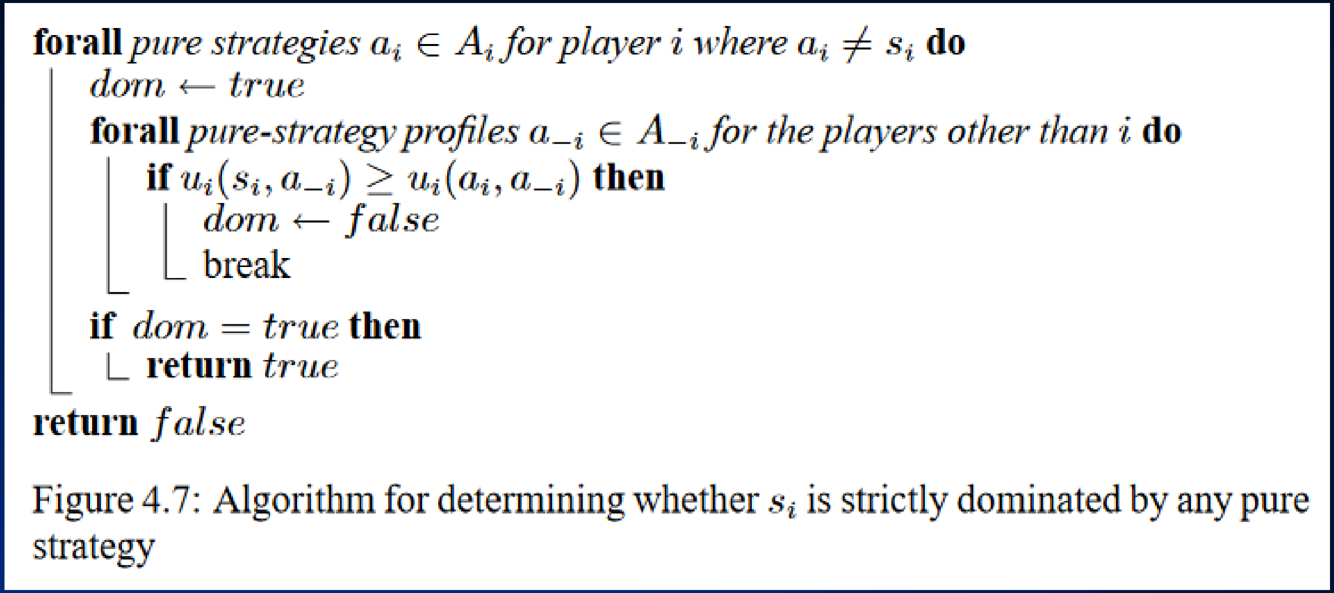
# Maxmin and minmax strategies for two-player general-sum games

- Let G be an arbitrary two-player game

  $G = (\{1,2\}, A_1 \text{ x } A_2, (u_1, u_2)).$
- Define the zero-sum game in which P1's utility is unchanged and P2's utility is the negative of P1's.

  $G' = (\{1,2\}, A_1 \text{ x } A_2, (u_1, -u_1)).$
- By Minmax Theorem every strategy for player 1 which is part of a Nash equilibrium strategy profile for G' is a maxmin strategy for player 1 in G'.
  - P1's maxmin strategy is independent of P2's utility function.
  - So P1's maxmin strategy is the same in G and G'.
- Same idea to compute minmax strategy for P2.

# Identifying dominated strategies

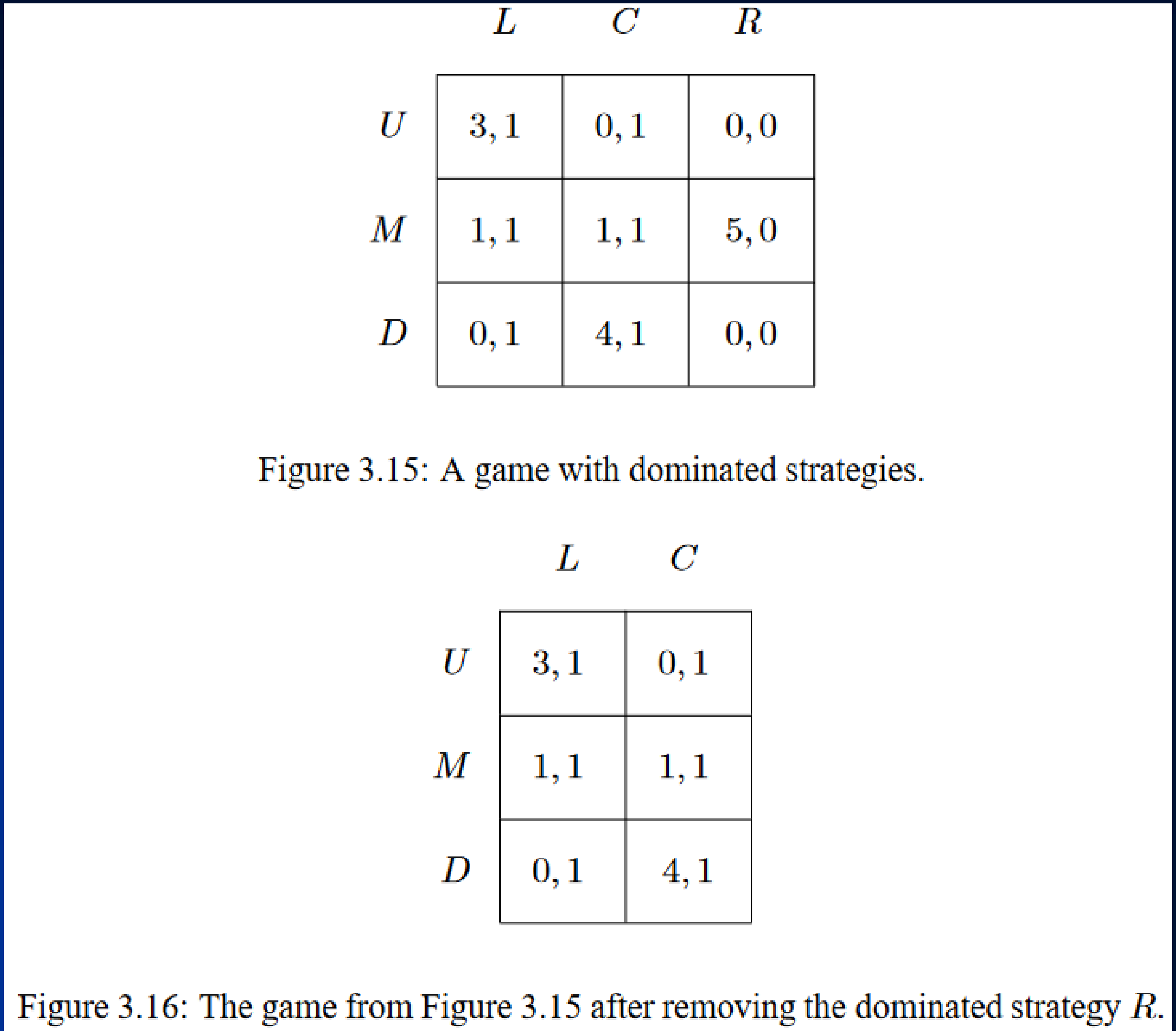


```
forall pure strategies $a_i \in A_i$ for player $i$ where $a_i \neq s_i$ do
    $dom \leftarrow true$
    forall pure-strategy profiles $a_{-i} \in A_{-i}$ for the players other than $i$ do
        if $u_i(s_i, a_{-i}) \geq u_i(a_i, a_{-i})$ then
            $dom \leftarrow false$
            break
    if $dom = true$ then
        return $true$
return $false$
```

Figure 4.7: Algorithm for determining whether $s_i$ is strictly dominated by any pure strategy

- Works because we do not need to check every *mixed*-strategy profile of the other players.
- If $u_i(s_i, a_{-i}) < u_i(a_i, a_{-i})$, for all $a_{-i}$ then there cannot exist any mixed-strategy profile $s_{-i}$ for which $u_i(s_i, s_{-i}) >= u_i(a_i, s_{-i})$, because of linearity of expectation.
- Can the algorithm be modified for weak dominance?

# Domination by a mixed strategy



| | $L$ | $C$ | $R$ |
|---|---|---|---|
| $U$ | $3, 1$ | $0, 1$ | $0, 0$ |
| $M$ | $1, 1$ | $1, 1$ | $5, 0$ |
| $D$ | $0, 1$ | $4, 1$ | $0, 0$ |

Figure 3.15: A game with dominated strategies.

| | $L$ | $C$ |
|---|---|---|
| $U$ | $3, 1$ | $0, 1$ |
| $M$ | $1, 1$ | $1, 1$ |
| $D$ | $0, 1$ | $4, 1$ |

Figure 3.16: The game from Figure 3.15 after removing the dominated strategy $R$.

- Strict domination

  Maximize ….

  Subject to $\Sigma_{j \text{ in } Ai}\, p_j\, u_i(a_j, a_{-i}) > u_i(s_i, a_{-i})$ for all $a_{-i}$ in $A_{-i}$

  $\Sigma_{j \text{ in } Ai}\, p_j = 1$

  $p_j >= 0$ for all j in $A_i$

- Valid?

# Domination by a mixed strategy

- Strict domination

Minimize $\Sigma_{j \text{ in } Ai}\, p_j$

Subject to $\Sigma_{j \text{ in } Ai}\, p_j\, u_i(a_j, a_{-i}) >= u_i(s_i, a_{-i})$ for all $a_{-i}$ in $A_{-i}$

$p_j >= 0$ for all j in $A_i$

- Weak domination

Maximize $\Sigma_{a\text{-}i \text{ in } A\text{-}i}\, [(\Sigma_{j \text{ in } Ai}\, p_j * u_i(a_j, a_{-i})) - u_i(s_i, a_{-i})$

Subject to $\Sigma_{j \text{ in } Ai}\, p_j\, u_i(a_j, a_{-i}) >= u_i(s_i, a_{-i})$ for all $a_{-i}$ in $A_{-i}$

$p_j >= 0$ for all j in $A_i$

$\Sigma_{j \text{ in } Ai}\, p_j = 1$

- It requires only polynomial time to iteratively remove dominated strategies until the game has been maximally reduced (i.e., no strategy is dominated for any player). A single step of this process consists of checking whether every pure strategy of every player is dominated by any other mixed strategy which requires us to solve at worst $\Sigma_{i \text{ in } N} |A_i|$ linear programs. Each step removes one pure strategy for each player, so there can be at most $\Sigma_{i \text{ in } N} (|A_i| - 1)$ steps.

Minimize $U^*_1$

Subject to $\Sigma_{k \text{ in } A2} u_1(a^j_1, a^k_2) * s^k_2 <= U^*_1$ for all j in $A_1$

$\Sigma_{k \text{ in } A2} s^k_2 = 1$

$s^k_2 >= 0$ for all k in $A_2$

Minimize $U^*_1$

Subject to $3 * s^1_2 + (-5) * s^2_2 + (-2) * s^3_2 <= U^*_1$

$1 * s^1_2 + 4 * s^2_2 + 1 * s^3_2 <= U^*_1$

$6 * s^1_2 + (-3) * s^2_2 + (-5) * s^3_2 <= U^*_1$

$s^1_2 + s^2_2 + s^3_2 = 1$

$s^1_2 >= 0, s^2_2 >= 0, s^3_2 >= 0$

# Linear programs

- A *linear program* is defined by:
  - a set of real-valued variables
  - a linear objective function (i.e., a weighted sum of the variables)
  - a set of linear constraints (i.e., the requirement that a weighted sum of the variables must be less than or equal to some constant).
- Let the set of variables be $\{x_1, x_2, \ldots, x_n\}$, which each $x_i$ in R. The objective function of a linear program, given a set of constraints $w_1, w_2, \ldots, w_n$, is

  Maximize $\Sigma^n_{i=1} w_i x_i$

- We can solve the dual linear program to obtain a Nash equilibrium strategy for player 1.

Maximize $U^*_1$

Subject to $\Sigma_{j \text{ in } A1} u_1(a^j_1, a^k_2) * s^j_1 >= U^*_1$ for all k in $A_2$

$\Sigma_{j \text{ in } A1} s^j_1 = 1$

$s^j_1 >= 0$ for all j in $A_1$

Minimize $U^*_1$

Subject to $\Sigma_{k \text{ in } A2}\, u_1(a^j_1, a^k_2) * s^k_2 <= U^*_1$ for all j in $A_1$

$\Sigma_{k \text{ in } A2}\, s^k_2 = 1$

$s^k_2 >= 0$ for all k in $A_2$

Minimize $U^*_1$

Subject to $3 * s^1_2 + (-5) * s^2_2 + (-2) * s^3_2 <= U^*_1$

$1 * s^1_2 + 4 * s^2_2 + 1 * s^3_2 <= U^*_1$

$6 * s^1_2 + (-3) * s^2_2 + (-5) * s^3_2 <= U^*_1$

$s^1_2 + s^2_2 + s^3_2 = 1$

$s^1_2 >= 0, s^2_2 >= 0, s^3_2 >= 0$

Maximize $U^*_1$

Subject to $\Sigma_{j \text{ in } A1} u_1(a^j_1, a^k_2) * s^j_1 >= U^*_1$ for all k in $A_2$

$\Sigma_{j \text{ in } A1} s^j_1 = 1$

$s^j_1 >= 0$ for all j in $A_1$

Maximize $U^*_1$

Subject to $a * s^1_2 + b * s^2_2 + c * s^3_2 <= U^*_1$

$d * s^1_2 + e * s^2_2 + f * s^3_2 <= U^*_1$

$g * s^1_2 + h * s^2_2 + j * s^3_2 <= U^*_1$

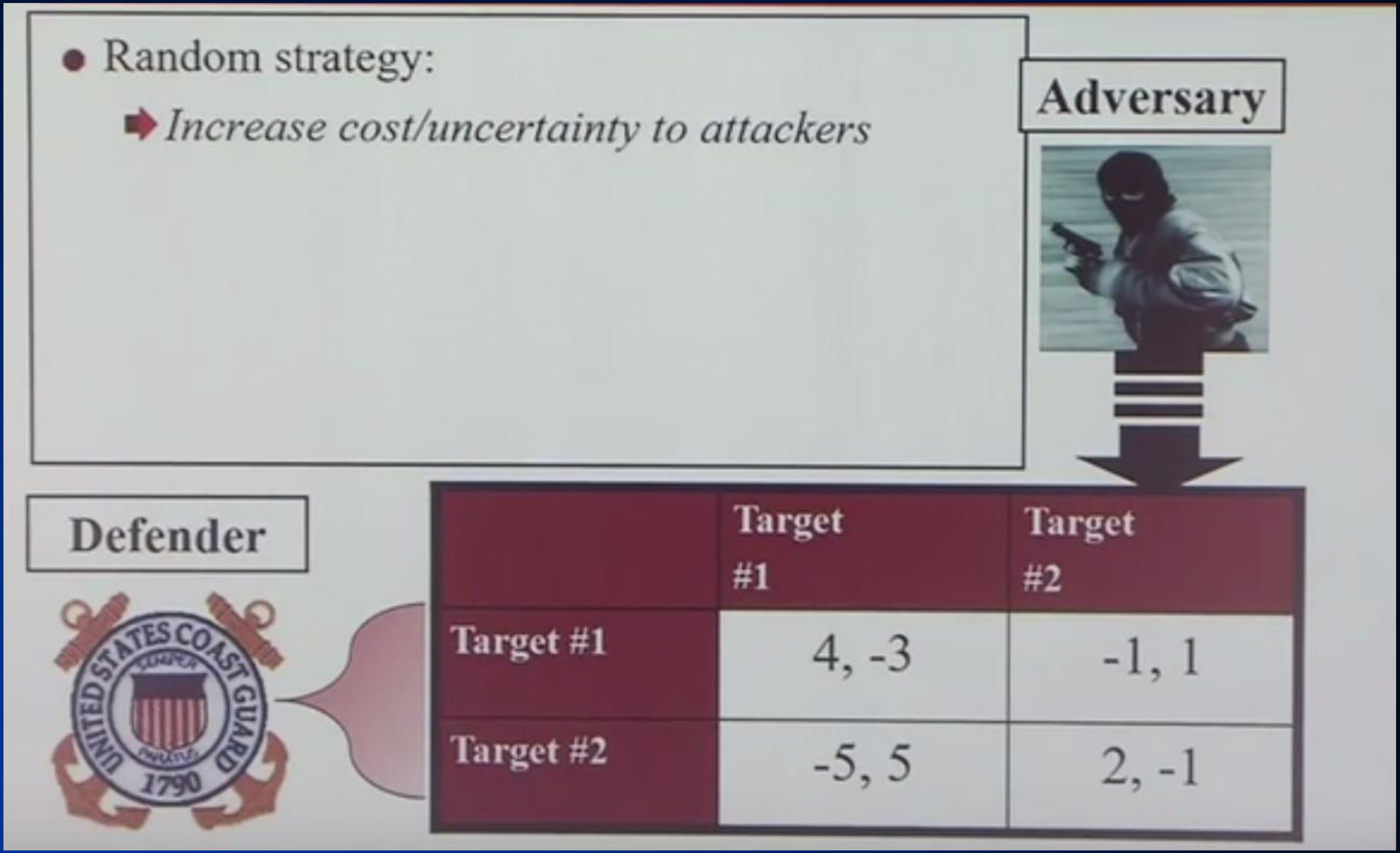
$s^1_2 + s^2_2 + s^3_2 = 1$

$s^1_2 >= 0, s^2_2 >= 0, s^3_2 >= 0$

# Why does this matter?

- Linear programs can be solved “efficiently.”
  - Ellipsoid method runs in polynomial time.
  - Simplex algorithm runs in worst-case exponential time, but runs efficiently in practice.

# Two-player general sum games



- Random strategy:
  - *Increase cost/uncertainty to attackers*

Adversary

Defender

|  | Target #1 | Target #2 |
|---|---|---|
| Target #1 | 4, -3 | -1, 1 |
| Target #2 | -5, 5 | 2, -1 |

- Minmax Theorem does not apply, so we cannot formulate as a linear program. We can instead formulate as a *Linear Complemetarity Problem (LCP)*.

Minimize ….. (No objective!)

Subject to $\Sigma_{k \text{ in } A2}\, u_1(a^j_1, a^k_2) * s^k_2 + r^j_1 = U^*_1$ for all j in $A_1$

$\Sigma_{j \text{ in } A1}\, u_2(a^j_1, a^k_2) * s^k_2 + r^k_2 = U^*_2$ for all k in $A_2$

$\Sigma_{j \text{ in } A1}\, s^j_1 = 1, \Sigma_{k \text{ in } A2}\, s^k_2 = 1$

$s^j_1, s^k_2 >= 0$ for all j in $A_1$, k in $A_2$

$r^j_1, r^k_2 >= 0$ for all j in $A_1$, k in $A_2$

$r^j_1 * s^j_1 = 0, r^j_2 * s^j_2 = 0$ for all j in $A_1$, k in $A_2$

- If we omitted the final constraints we would still have an LP, but we would have a flaw. The variables $U^*_1$ and $U^*_2$ would be allowed to take unboundedly large values, because all of these constraints remain satisfied when both $U^*_i$ and $r^*_i$ are increased by the same constant, for any given i and j.
- The new constraint ensures that whenever an action is played by a given player with positive probability then the corresponding slack variable must be zero. Under this requirement, each slack variable can be viewed as the player's incentive to deviate from the corresponding action.

# Polynomial time (P)

- An algorithm is said to be of **polynomial time** if its running time is upper bounded by a polynomial expression in the size of the input for the algorithm, i.e., $T(n) = O(n^k)$ for some constant $k$. Problems for which a deterministic polynomial time algorithm exists belong to the complexity class **P**, which is central in the field of computational complexity theory. Cobham's thesis states that polynomial time is a synonym for "tractable", "feasible", "efficient", or "fast".

- Logarithmic time: Binary Search
- Linearithmic time (O nlogn): MergeSort
- Linear time (O(n)): finding smallest or largest item in an unsorted array.
- Quadratic time: bubble sort, insertion sort
  - **Bubble sort** is a simple sorting algorithm that repeatedly steps through the list to be sorted, compares each pair of adjacent items and swaps them if they are in the wrong order. The pass through the list is repeated until no swaps are needed, which indicates that the list is sorted. The algorithm, which is a comparison sort, is named for the way smaller or larger elements "bubble" to the top of the list.
- Cubic time: Naive multiplication of two *n*×*n* matrices.

# P vs. NP

- John Nash letter to NSA: https://www.nsa.gov/news-features/declassified-documents/nash-letters/assets/files/nash_letters1.pdf

- Informally, **NP** is the set of all decision problems for which the instances where the answer is "yes" have efficiently *verifiable* proofs. More precisely, these proofs have to be verifiable by deterministic computations that can be performed in polynomial time.

- Computing a Nash equilibrium in multiplayer (or two-player non zero-sum games): “a most fundamental computational problem whose complexity is wide open” and “together with factoring, the most important concrete open question on the boundary of P today”

# PPAD

- In computer science, **PPAD** (*"Polynomial Parity Arguments on Directed graphs"*) is a complexity class introduced by Christos Papadimitriou in 1994. PPAD is a subclass of TFNP based on functions that can be shown to be total by a parity argument.
- PPAD is a class of problems that are believed to be hard, but obtaining PPAD-completeness is a weaker evidence of intractability than that of obtaining NP-completeness. PPAD problems cannot be NP-complete, for the technical reason that NP is a class of decision problems, but the answer of PPAD problems is always yes, as a solution is known to exist, even though it might be hard to find that solution.
  - By Nash's Theorem, we know a Nash equilibrium always exists.

# Hardness

- **NP-hardness** (*n*on-deterministic *p*olynomial-time hard), in computational complexity theory, is a class of problems that are, informally, "at least as hard as the hardest problems in NP." More precisely, a problem *H* is NP-hard when every problem *L* in NP can be reduced in polynomial time to *H*, that is: assuming a solution for *H* takes 1 unit time, we can use *H*'s solution to solve *L* in polynomial time. As a consequence, finding a polynomial algorithm to solve any NP-hard problem would give polynomial algorithms for all the problems in NP, which is unlikely as many of them are considered hard.
- NP-completeness: in NP and NP-hard.

- Polynomial time:
  - Computing Nash equilibrium in two-player zero-sum strategic-form games
  - Computing Minmax and Maxmin values in two-player general-sum games
  - Computing Subgame Perfect Nash equilibrium in two-player zero-sum perfect-information games
  - Computing Nash equilibrium in two-player zero-sum extensive-form games
  - Various forms of domination

- NP-hard:
  - strategy elimination, reduction identity, uniqueness and reduction size problems for iterated weak dominance
  - Finding Nash equilibrium that maximizes “social welfare,” or satisfies other properties https://users.cs.duke.edu/~conitzer/nashGEB08.pdf
- PPAD-hard:
  - Computing Nash equilibrium in two-player general-sum games, or in games with >= 2 players, for both strategic-form and extensive-form games

- B. von Stengel (2002), Computing equilibria for two-person games. Chapter 45, *Handbook of Game Theory, Vol. 3*, eds. R. J. Aumann and S. Hart, North-Holland, Amsterdam, 1723-1759.
  - http://www.maths.lse.ac.uk/personal/stengel/TEXTE/nashsurvey.pdf
- Longer earlier version (with more details on equivalent definitions of degeneracy, among other aspects):
  B. von Stengel (1996), Computing Equilibria for Two-Person Games. Technical Report 253, Department of Computer Science, ETH Zürich.

- Define E = [1,…,1], e = 1, F = [1,…,1], f = 1
- Given a fixed y in Y, a best response of player 1 to y is a vector x in X that maximizes the expression $x^T(Ay)$. That is, x is a solution to the LP:

  Maximize $x^T(Ay)$

  Subject to Ex = e, x >= 0

- The dual of this LP with variables u:

  Minimize $e^Tu$

  Subject to $E^Tu$ >= Ay

- So a minmax strategy y of player 2 (minimizing the maximum amount she has to pay) is a solution to the LP

  Minimize $e^T u$

  Subject to $Fy = f$

  $E^T u - Ay >= 0$

  $y >= 0$

- Dual LP:

  Maximize $f^T v$

  Subject to $Ex = e$

  $F^T v - B^T x <= 0$

  $x >= 0$

- Theorem: The game (A,B) has the Nash equilibrium (x,y) if and only if for suitable u,v

$$Ex = e$$

$$Fy = f$$

$$E^T u - Ay >= 0$$

$$F^T v - B^T x >= 0$$

$$x, y >= 0$$

- This is called a *linear complementarity program*.
- Best algorithm is Lemke Howson Algorithm.
  - Does NOT run in polynomial time. Worst-case exponential.
- Computing a Nash equilibrium in these games is PPAD-complete, unlike for two-player zero-sum games where it can be done in polynomial time.

- Assume disjoint strategy sets M and N for both players. Any mixed strategy x in X and y in Y is labeled with certain elements if M union N. These labels denote the unplayed pure strategies of the player and the pure best responses of his or her opponent. For i in M and j in N, let
  - X(i) = {x in X| $x_i$ = 0},
  - X(j) = {x in X| $b_j$x >= $b_k$x for all k in N}
  - Y(i) = {y in Y | $a_i$y >= $a_k$y for all k in M}
  - Y(j) = {y in Y | $y_j$ = 0}
- Then x has label k if x in X(k) (i.e., x is a best response to strategy k for player 2), and y has label k if y in Y(k), for k in M Union N.

- Theorem: The game (A,B) has the Nash equilibrium (x,y) if and only if for suitable u,v

$$Ex = e$$

$$Fy = f$$

$$E^T u - Ay >= 0$$

$$F^T v - B^T x >= 0$$

$$x, y >= 0$$

- *Complementarity condition:* requires that whenever an action is played by a given player with positive probability (i.e., whenever an action is in the support of a given player's mixed strategy), then the corresponding slack variable must be zero. Under this requirement, each slack variable can be viewed as the player's incentive to deviate from the corresponding action. Thus, the complementarity condition captures the fact that, in equilibrium, all strategies that are played with positive probability must yield the same expected payoff, while all strategies that lead to lower expected payoffs are not played.

- Clearly, the best-response regions X(j) for j in N are polytopes whose union is X. Similarly, Y is the union of the sets Y(i) for i in M. Then a Nash equilibrium is a *completely labeled* pair (x,y) since then by Theorem 2.1, any pure strategy k of a player is either a best response or played with probability zero, so it appears as a label of x or y.

- Theorem: A mixed strategy pair (x,y) in X x Y is a Nash equilibrium of (A,B) if and only if for all k in M Union N either x in X(k) or y in Y(k) (or both).

- For the following game, the labels of X and Y are:

| | **L** | **R** |
|---|---|---|
| T | 0, 1 | 6, 0 |
| M | 2, 0 | 5, 2 |
| B | 3, 4 | 3, 3 |

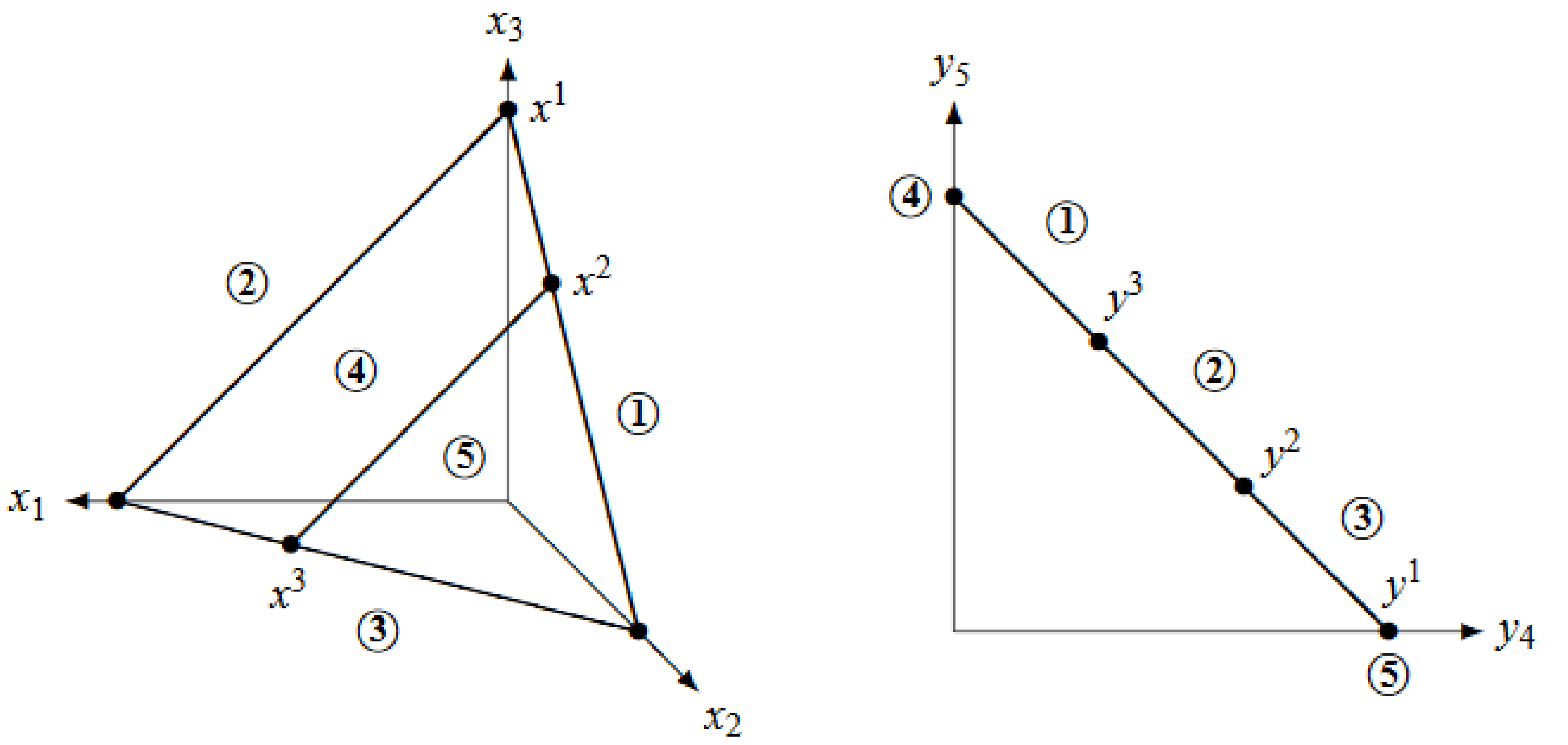


Figure 2.2. Mixed strategy sets $X$ and $Y$ of the players for the bimatrix game $(A,B)$ in (2.15). The labels $1,2,3$, drawn as circled numbers, are the pure strategies of player 1 and marked in $X$ where they have probability zero, in $Y$ where they are best responses. The pure strategies of player 2 are similar labels $4,5$. The dots mark points $x$ and $y$ with a maximum number of labels.

- The equilibria are:
  - $(x_1,y_1) = ((0,0,1),(1,0))$, where $x_1$ has the labels 1, 2, 4 (and $y_1$ has the remaining labels 3 and 5),
  - $(x_2,y_2) = ((0,1/3,2/3),(2/3,1/3))$, with labels 1, 4, 5 for $x_2$
  - $(x_3,y_3) = ((2/3,1/3,0),(1/3,2/3))$, with labels 3, 4, 5 for $x_3$

- This “inspection” is effective at finding equilibria of games of size up to 3x3. It works by inspecting any point x for P1 with m labels and checking if there is a point y having the remaining n labels. A game is “nondegenerate” if any x has at most m labels and every y has at most n labels.
- “Most” games are nondegenerate, since having an additional label imposes an additional equation that will usually reduce the dimension of the set of points having these labels by one. Since the complete set X has dimension m-1, we expect no points to have more than m labels. This will fail only in exceptional cirtcumstances if there is a special relationship between the elements of A and B.

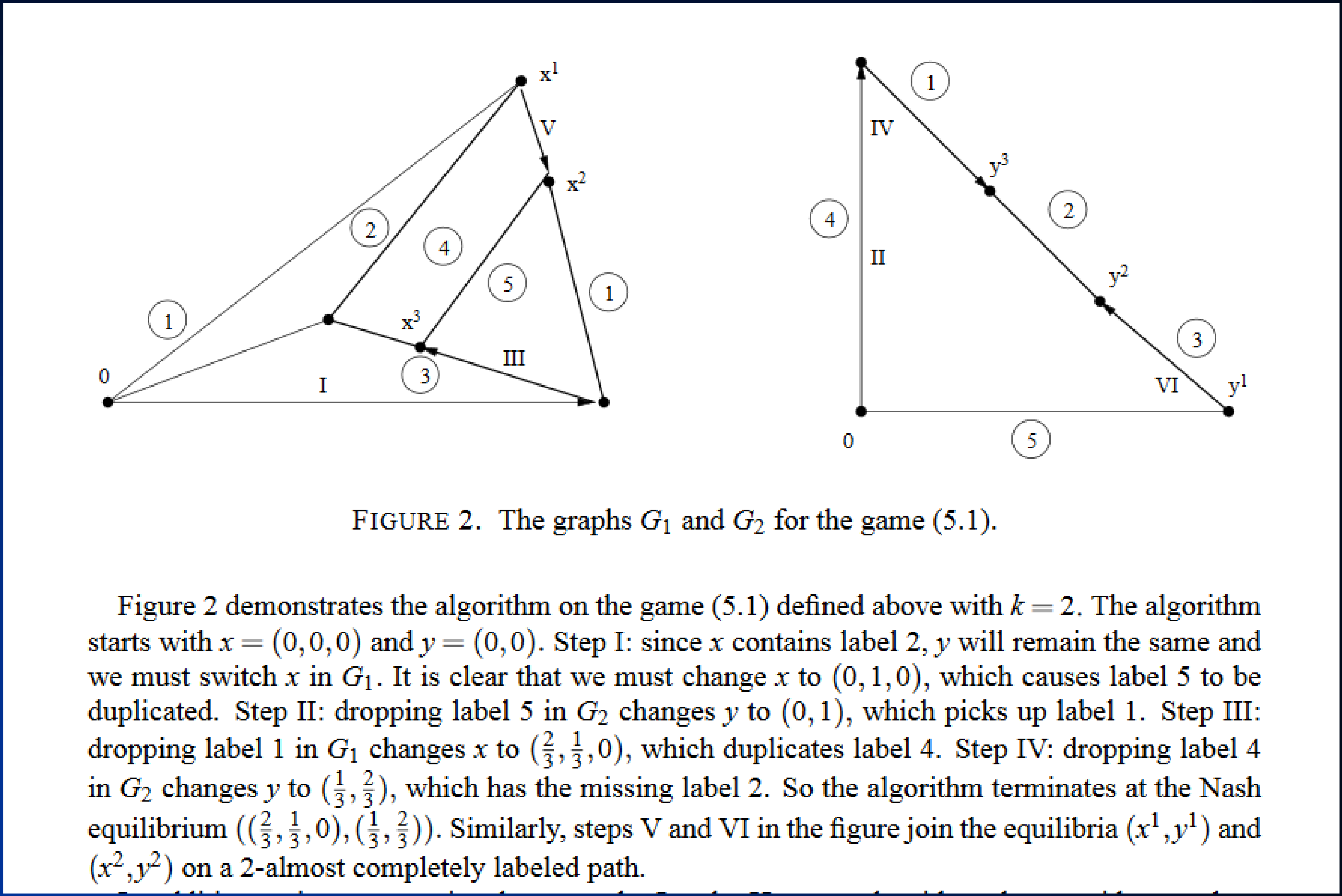


FIGURE 2. The graphs $G_1$ and $G_2$ for the game (5.1).

Figure 2 demonstrates the algorithm on the game (5.1) defined above with $k = 2$. The algorithm starts with $x = (0,0,0)$ and $y = (0,0)$. Step I: since $x$ contains label 2, $y$ will remain the same and we must switch $x$ in $G_1$. It is clear that we must change $x$ to $(0,1,0)$, which causes label 5 to be duplicated. Step II: dropping label 5 in $G_2$ changes $y$ to $(0,1)$, which picks up label 1. Step III: dropping label 1 in $G_1$ changes $x$ to $(\frac{2}{3},\frac{1}{3},0)$, which duplicates label 4. Step IV: dropping label 4 in $G_2$ changes $y$ to $(\frac{1}{3},\frac{2}{3})$, which has the missing label 2. So the algorithm terminates at the Nash equilibrium $((\frac{2}{3},\frac{1}{3},0),(\frac{1}{3},\frac{2}{3}))$. Similarly, steps V and VI in the figure join the equilibria $(x^1,y^1)$ and $(x^2,y^2)$ on a 2-almost completely labeled path.

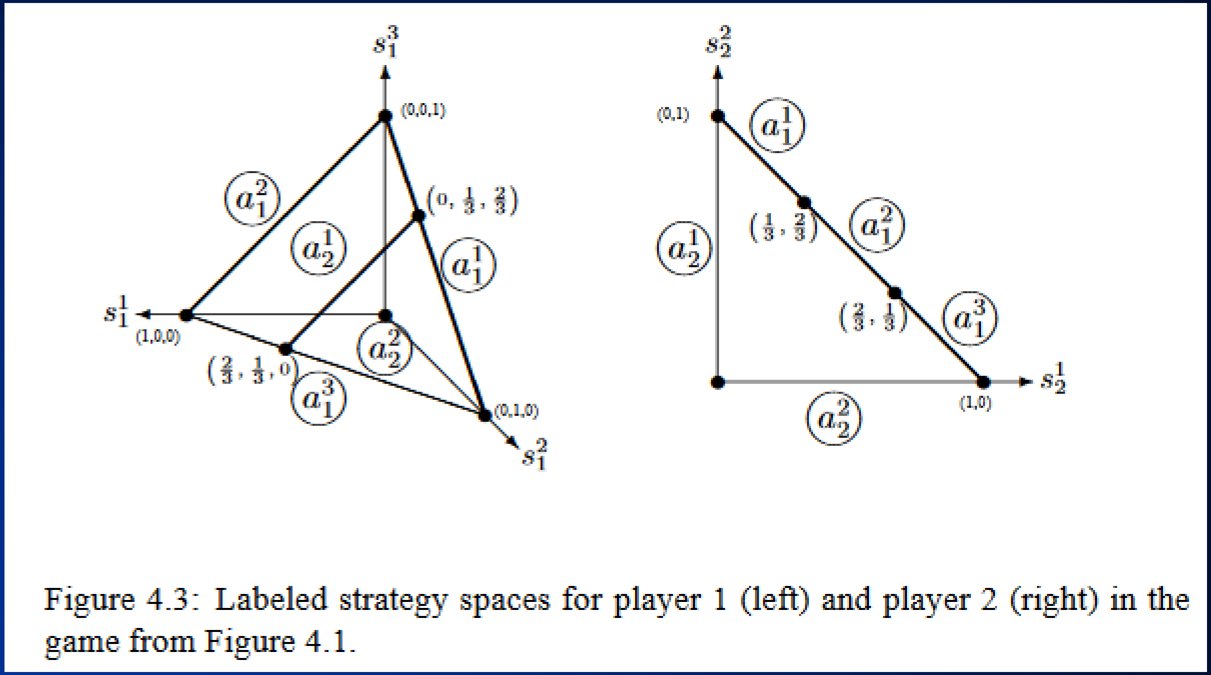
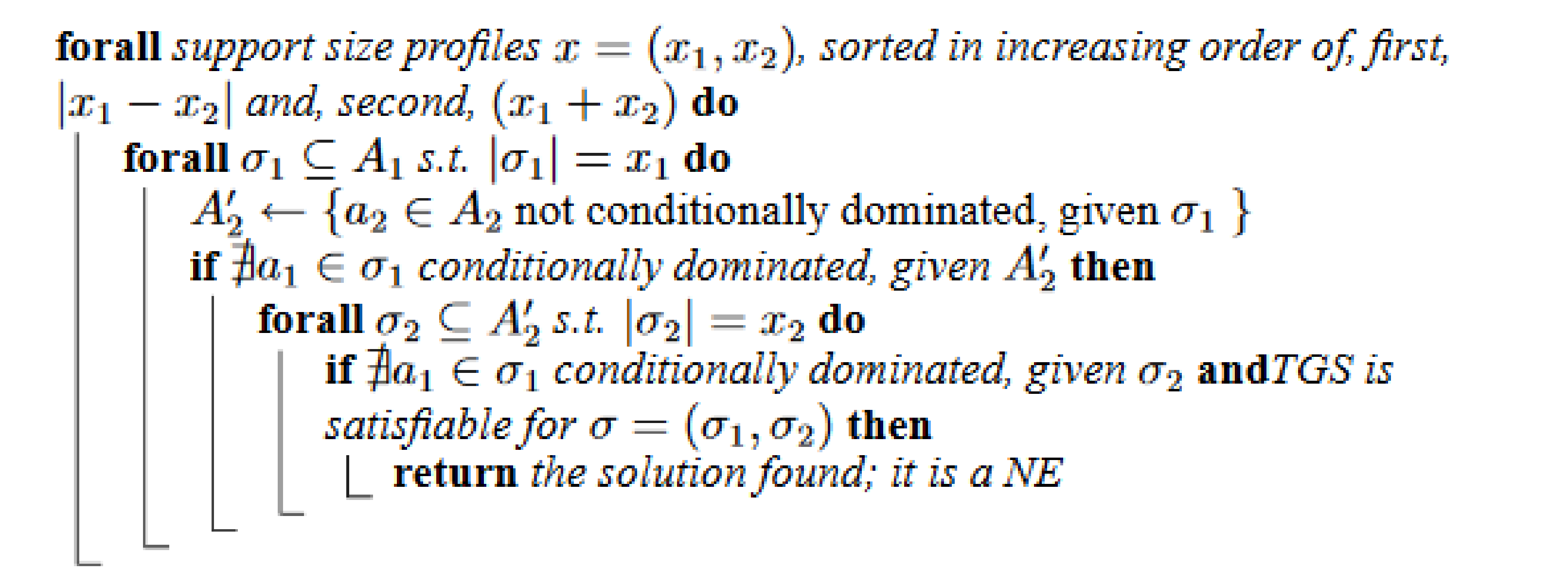


Figure 4.3: Labeled strategy spaces for player 1 (left) and player 2 (right) in the game from Figure 4.1.

# Support enumeration algorithm



```
forall support size profiles $x = (x_1, x_2)$, sorted in increasing order of, first,
$|x_1 - x_2|$ and, second, $(x_1 + x_2)$ do
    forall $\sigma_1 \subseteq A_1$ s.t. $|\sigma_1| = x_1$ do
        $A'_2 \leftarrow \{a_2 \in A_2$ not conditionally dominated, given $\sigma_1$ $\}$
        if $\nexists a_1 \in \sigma_1$ conditionally dominated, given $A'_2$ then
            forall $\sigma_2 \subseteq A'_2$ s.t. $|\sigma_2| = x_2$ do
                if $\nexists a_1 \in \sigma_1$ conditionally dominated, given $\sigma_2$ andTGS is
                satisfiable for $\sigma = (\sigma_1, \sigma_2)$ then
                    return the solution found; it is a NE
```

Figure 4.6: The SEM algorithm

# n-player general-sum games

- For n-player games with n >= 3, the problem of computing an NE can no longer be expressed as an LCP. While it can be expressed as a *nonlinear complementarity problem*, such problems are often hopelessly impractical to solve exactly.
- Can solve sequence of LCPs (generalization of Newton's method).
  - Not globally convergent
- Formulate as constrained optimization (minimization of a function), but also not globally convergent (e.g., hill climbing, simulated annealing can get stuck in local optimum)
- Simplicial subdivision algorithm (Scarf)
  - Divide space into small regions and search separately over the regions.
- Homotopy method (Govindan and Wilson)
  - n-player extension of Lemke-Howson Algorithm

# Next class: refinements of Nash equilibrium

- Is it too strict?
  - Does not exist in all games
  - Might rule out some more “reasonable” strategies
- Not strict enough?
  - Potentially many equilibria to select through
- Just right?

# Next class

- Brief introduction to Game Theory Explorer.
- Go over HW1.
- Refinements of Nash equilibrium.

# Assignment

- HW2 due 2/21.
- Reading for next class: chapter 7 from main textbook.