Diagonal Games

Definition: A diagonal game is a game such that its payoff matrix is a diagonal matrix.

Solution for diagonal games:

Given $\begin{pmatrix} d_1 & & \\ & \ddots & \\ & & d_n \end{pmatrix}$

(i) d_i are all positive or all negative.

Then, $$\left(\frac{1/d_1}{1/d_1+\cdots+1/d_n},\cdots,\frac{1/d_n}{1/d_1+\cdots+1/d_n}\right)$$

is an equalizing strategy for each player.

(ii) Otherwise, 0 is a saddle point.

Examples:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 7 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 \\ 0 & -5 & 0 \\ 0 & 0 & 7 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 7 \end{pmatrix}$$

Answer: $V=(1/1 + 1/5 + 1/7)^{-1}$

Optimal strategy:

$<(1/V,\ 1/5V, 1/7V), (1/V, 1/5V, 1/7V)>$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 7 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & -5 & 0 \\ 0 & 0 & 7 \end{pmatrix}$$

Answer: V=0 achieved at saddle points

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & -5 & 0 \\ 0 & 0 & 7 \end{pmatrix}$$

Finding safety (Optimal) strategies when the value is given

Let V be the value of the game. Then, p~ is called a safety (optimal) strategy of Player I, whenever

$$\text{Min}_q\ (p\ \sim)^T Aq = V = \text{Max}_p\ \text{Min}_q\ p^T Aq$$

Similary, q~ is called a safety (optimal) strategy of Player II, whenever

$$\text{Max}_p\ p^T Aq\sim\ = V = \text{Min}_q\ \text{Max}_p\ p^T Aq$$

$\text{Min}_q \; p^T A q = V$ can be expressed as a system of n inequalities by noting that for a fixed p, the minimum of $p^T A q$ is attained at a pure strategy q.

Theorem: Let $A=(a_{ij})$ be a matrix game with value V. The p is a safety strategy for Player I iff

The following n inequalities are satisfied.

$$p_1 a_{11} + \ldots + p_m a_{m1} \geq V$$

$$\vdots \qquad \vdots \qquad \qquad \vdots$$

$$p_1 a_{1n} + \ldots + p_m a_{mn} \geq V$$

Proof: (Necessity): If p is a safety strategy then

$\text{Min}_q\ p^T A q = V$, where q is any mixed strategy of Player II. In particular for the n pure strategies

$e_1, \ldots, e_n$ we have

$p^T A e_1 \geq V, \ldots, p^T A e_n \geq V$, which are exactly the inequalities in the statement of the Theorem.

(Sufficiency): Suppose the n inequalities in the statement of the Theorem are satisfied. Then,

$\text{Min}_q\ p^T A q \geq V$. Note that we must have

$\text{Min}_q\ p^T A q = V$. Otherwise, $\text{Min}_q\ p^T A q > V$ will lead to value of the game is greater than V. Hence, p is a safety strategy.

Similarly, $\mathrm{Max}_p\, p^T A q = V$ can be expressed as a system of m inequalities by noting that for a fixed q, the maximum of $p^T A q$ is attained at a pure strategy p.

Theorem: Let $A=(a_{ij})$ be a matrix game with value V. The q is a safety strategy for Player II iff

The following m inequalities are satisfied.

$q_1 a_{11} + \ldots + q_n a_{1n} \le V$

$\vdots \qquad\qquad \vdots \qquad \vdots$

$q_1 a_{m1} + \ldots + q_n a_{mn} \le V$

Symmetric Games:

A game is symmetric if the rules do not distinguish between players. Both players have the same options, and the payoff when Player I uses i and Player II uses j is the negative of the payoff when Player I uses j and Player II uses i.

Thus the payoff matrix can be arranged to be skew-symmetric, i.e. $A= -A^T$.

We will show that the value of a symmetric game is 0. Hence, we can find its safety strategies by solving a system of inequalities.

Example: Scissor-Rock-Paper

Theorem: A finite symmetric game has value zero. Any strategy optimal (safety) for one player is also optimal for the other.

Proof: Let p be an optimal strategy for Player I. If Player II uses p. Then, $p^TAp=0$ because $p^TAp=(p^TAp)^T=-p^TAp$.

Thus, $V_{-} \leqslant 0$. Arguing from the optimal strategy of Player II, we also see that $V^{-} \geqslant 0$. Therefore, $V=V_{-}=V^{-}=0$ as claimed.

Let $p^{\sim}$ be an optimal strategy for Player I.

Then, $\mathrm{Min}_q p^{\sim T}Aq=V=0$.

Since, $0= - \mathrm{Min}_q p^{\sim T}Aq= \mathrm{Max}_q(-p^{\sim T}Aq)$

$=\mathrm{Max}_q(p^{\sim T}(-A)q)$

$= \mathrm{Max}_q(p^{\sim T}A^T q)$

$= \mathrm{Max}_q(q^T A\, p^{\sim})$

$p^{\sim}$ is then also optimal for Player II.

Example: Solve the following symmetric game.

$$\begin{pmatrix} 0 & -1 & 2 \\ 1 & 0 & -1 \\ -2 & 1 & 0 \end{pmatrix}$$

Solution: Let (p_1,p_2,p_3) be the optimal strategy. Then,

$$\begin{aligned}
p_1 \cdot 0 &+ p_2 \cdot 1 &+ p_3 \cdot (-2) &\geq 0 \\
p_1 \cdot (-1) &+ p_2 \cdot 0 &+ p_3 \cdot 1 &\geq 0 \\
p_1 \cdot 2 &+ p_2 \cdot (-1) &+ p_3 \cdot 0 &\geq 0
\end{aligned}$$

$$\begin{aligned}
p_2 &\geq 2p_3 \\
p_3 &\geq p_1 \\
2p_1 &\geq p_2
\end{aligned}$$

Thus,$2p_1 \geqslant p_2 \geqslant 2p_3$, $p_3 \geqslant p_1$. $p_1=p_2/2=p_3$.

Since $p_1+p_2+p_3=1$, $p_1=p_3=1/4$, $p_2=1/2$.

Algorithms to find optimal strategies

Now, we know that if the value of a game is known, we can find the sets of optimal strategies by solving systems of inequalities.

Question: Find effective algorithms to solve 2-person 0-sum games?

Iterative solution of games by fictious play

- Step 1: Player I chooses a pure strategy p_1. Set $x_1 = p_1$. Player II chooses a pure strategy BR to x_1. Call it y_1.
- Step 2: Player I, given the history of II's play, counter a pure strategy BR to y_1. Call it x_2. Player II,given history of I's play, counter with a pure strategy BR to $(x_1 + x_2)/2$. Call it y_2.
- Step k+1: Player I, given the history of II's play, counter a pure strategy BR to $(y_1+\ldots+y_k)/k$. Call it x_{k+1} Player II, given the history of I's play, counter with a pure strategy BR to $(x_1+\ldots+x_{k+1})/k+1$. Call it y_{k+1}.
- Julia Robinson proved in 1951 that iterative procedures of this type must converge to the optimal strategies.

Example:

$Game\ \#$	$PlayerI$	$PlayerII$	$R1$	$R2$	$C1$	$C2$
1	$R2$	$C1$	0	1	1	0
2	$R1$	$C2$	1	1	2	0
3	$R1$	$C2$	2	1	2	1
4	$R2$	$C2$	2	2	2	2
5	$R2$	$C2$	2	3	2	3
6	$R2$	$C2$	2	4	2	4
7	$R2$	$C1$	2	5	3	4
8	$R2$	$C1$	2	6	4	4
9	$R2$	$C1$	2	7	5	4
10	$R2$	$C1$	2	8	6	4
11	$R2$	$C1$	2	9	7	4
12	$R2$	$C1$	2	10	8	4

	$C1$	$C2$
$R1$	0	-1
$R2$	$-\frac{1}{2}$	0

Solving 2-Person 0-sum games by linear programming

- Linear Programming:
- The basic problem of linear programming, determining the optimal value of a linear function subject to linear constraints, arises in a wide variety of situations.
- In 1939 the Russian mathematician L.V. Kantorovich published a monograph entitled “Mathematical Methods in the Organization and Planning of Production”. Kantorovich went unrecognized in the West.
- Through the works of Frank Hitchcock and George Stigler and their efforts in World War II, it became clear that a feasible method for solving linear programming problems was needed. Then in 1947 George Dantzig developed the simplex method. The first proof of the Duality Theorem was by Gale, Kuhn and Tucker. John von Neumann immediately recognized the importance of the concept of duality
- Simplex method: The simplex method gives an algorithm to go from one vertex to another vertex in searching for the optimal value.

Example of a linear programming problem:

In order to ensure optimal health, a lab technician needs to feed the rabbits a daily diet containing a minimum of 24 grams (g) of fat, 36 g of carbohydrates, and 4 g of protein. But the rabbits should be fed no more than five ounces of food a day. Rather than order rabbit food that is custom-blended, it is cheaper to order Food X and Food Y, and blend them for an optimal mix. Food X contains 8 g of fat, 12 g of carbohydrates, and 2 g of protein per ounce, and costs $0.20 per ounce. Food Y contains 12 g of fat, 12 g of carbohydrates, and 1 g of protein per ounce, at a cost of $0.30 per ounce.

What is the optimal blend?

x: number of ounces of Food X
y: number of ounces of Food Y

Minimize $C = 0.2x + 0.3y$

(fat)	$8x + 12y \geq 24$
(carbs)	$12x + 12y \geq 36$
(protein)	$2x + 1y \geq 4$
(weight)	$x + y \leq 5$
	$x \geq 0,\ y \geq 0$

Reduction of a 2-person 0-sum game to a Linear Programming Problem.

A Linear Programming Problem is defined as the problem of choosing real variables to maximize or minimize a linear function of the variables, called the objective function, subject to linear constraints on the variables. The constraints may be equalities or inequalities.

The Primal Problem of a linear programming problem is in the following form.

Choose $x_1,\ldots,x_{m,}$ to

minimize $c_1x_1 + \ldots + c_mx_m$

subject to the constraints

$$
\begin{array}{lcl}
a_{11}x_1 + \ldots + a_{m1}x_m & \geqslant & b_1 \\
\vdots \qquad\qquad \vdots & & \vdots \\
a_{1n}x_1 + \ldots + a_{mn}x_m & \geqslant & b_n
\end{array}
$$

and $x_i \geqslant 0$ for $i = 1,\ldots,m$

The Dual Problem to the primal problem is to

choose $y_1,\ldots,y_n$ to

maximize $b_1y_1+\ldots+b_ny_n$

subject to the constraints

$$\begin{array}{l} a_{11}y_1+\ldots+a_{1n}y_n \leqslant c_1 \\ \vdots \qquad\qquad\quad \vdots \qquad\quad \vdots \\ a_{m1}y_1+\ldots+a_{mn}y_n \leqslant c_m \end{array}$$

and $y_j \geqslant 0$ for $j = 1,\ldots,n$

Theorem: (Duality Theorem of LP) The optimal value of the Primal Problem and the optimal value of the Dual Problem are equal.

The Duality Theorem implies the Minimax Theorem in game theory.

Let us consider the game problem from Player I's point of view. He wants to choose

$p=(p_1, \ldots, p_m)$ to maximize $\mathrm{Min}_q p^T A q$

subject to the constraint $p \in X^*$.

This becomes the mathematical program:

Choose $p_1, \ldots, p_m$ to

maximize $[\min_j (p_1 a_{1j}+\ldots+p_m a_{mj})]$

subject to the constraints $p_1 + \cdot \cdot \cdot + p_m = 1$

and $p_i \geq 0$ fo $i = 1, \ldots, m$.

Although the constraints are linear, the objective function is not a linear function of the p's because of the min operator, so this is not a linear program. However, it can be changed into a linear program through a trick.

Trick:

Add one new variable v to Player I's list of variables!

Restrict v to be less than the objective function,

$v \leq \min_j (p_1 a_{1j} + \ldots + p_m a_{mj})$.

Then, make v as large as possible subject to this new constraint.

The problem becomes:

Choose v and $p_1,\ldots,p_m$ to

maximize v

subject to the constraints

$$
\begin{array}{l}
v \leqslant p_1 a_{11} + \ldots + p_m a_{m1} \\
\vdots \\
v \leqslant p_1 a_{1n} + \ldots + p_m a_{mn}
\end{array}
$$

$p_1+\ldots+p_m=1$ and $p_i \geqslant 0$ for $i=1,..m$

The optimal value of the above problem is V _ the lower value of the game.

This is still not good enough. Is v nonegative?

We will transform the above optimization problem into a linear programming problem so that we can use the theory of linear programming to find the value of a game.

By adding a large constant to the game matrix if necessary, we may assume the optimal value for is positive. Now we can express the problem as:

maximize v

subject to the constraints

$$
\begin{array}{l}
1 \leqslant (p_1/v)a_{11} + \ldots + (p_m/v)a_{m1} \\
\vdots \qquad \vdots \qquad\qquad\qquad \vdots \\
1 \leqslant (p_1/v)a_{1n} + \ldots + (p_m/v)a_{mn}
\end{array}
$$

Let $x_1 = (p_1/v), \ldots x_m = (p_m/v)$.

Then, $x_1 + \ldots + x_m = 1/v$ and $x_i \geqslant 0$ for $i = 1, \ldots m$

Then, it is equivalent to study the following optimization problem.

minimize $(x_1+\ldots+x_m)$

subject to the constraints

$$1 \leqslant x_1 a_{11} + \ldots + x_m a_{m1}$$
$$\vdots \qquad \vdots \qquad \qquad \vdots$$
$$1 \leqslant x_1 a_{1n} + \ldots + x_m a_{mn}$$

$x_i \geqslant 0$ for $i = 1,\ldots,m$

This is the Primal Problem of a linear programming problem. The optimal value of this problem is $1/V$ _

The Dual Problem for the above problem is in the following.

maximize $(y_1+\ldots+y_n)$
subject to the constraints

$$
\begin{array}{l}
1 \geqslant y_1 a_{11} + \ldots + y_n a_{1n} \\
\vdots \qquad \vdots \qquad\qquad \vdots \\
1 \geqslant y_1 a_{m1} + \ldots + y_n a_{mn}
\end{array}
$$

$y_j \geqslant 0$ for $j = 1,\ldots,n$

The optimal value of this problem is $1/V^-$.

The Duality Theorem in linear programming then implies

$$\underline{V} = \overline{V},$$

the Minimax Theorem in Game Theory!

Linear Programming problems can be solved by the Simplex Algorithm developed by George Dantzig.

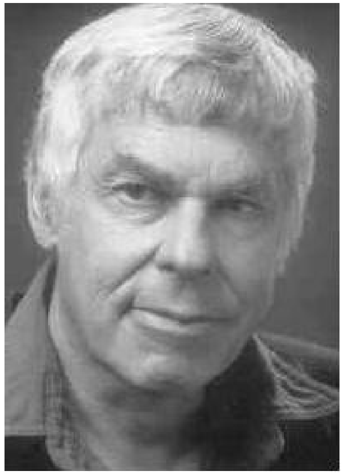
How good (fast) is the Simplex Algorithm?

Lovasz: “ If one would take statistics about which mathematical problem is using up most of the computer time in the world, then (not counting database handling problems like sorting and searching) the answer would probably be linear programming.”

The Simplex Algorithm performs efficiently on most LP problems but it performs badly on some LP problems (first example given by Klee and Minty).

However, these are just exceptional cases.

Steven Smale proved in 1983 that the "average steps" needed in the Simplex Algorithm grows slower than any prescribed root of the number of variables.



Polynomial time algorithm

- In 1979, the Russian mathematician Leonid Khatchian announced a polynomial time algorithm for the resolution of the linear programming problem. The algorithm uses a sequence of ellipsoids to drive to a solution of a linear programming problem.
- A polynomial time linear programming algorithm using an interior point method was found by Karmarkar (1984). Arguably, interior point methods were known as early as The media hype accompanying Karmarkar's announcement led to these methods receiving a great deal of attention. However, it should be noted that while Karmarkar claimed that his implementation was much more efficient than the simplex method, the potential of interior point method was established only later. By 1994, there were more than 1300 published papers on interior point methods.

Game Theory provides a framework and a formal language for dealing with “rules of the game” such as the process (offers, counteroffers, ploys or bluffs), the information patterns (“who knows, what, when”) and all other details.

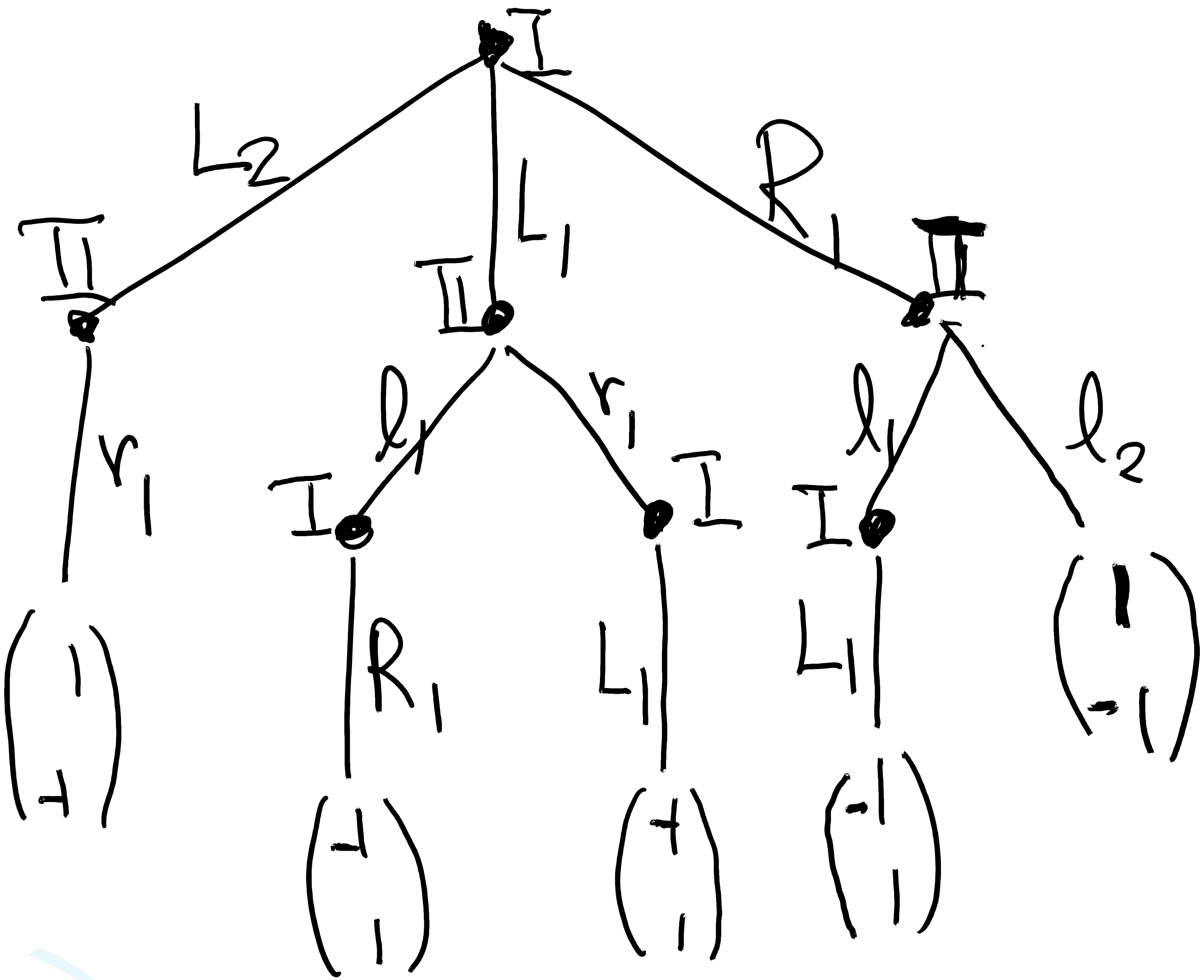
Games in Extensive Form

Goal: Give a COMPLETE mathematical description of the game

Example:
(Last Year at Marienbad)

Consider a two-pile Misère Nim Game:

- The pile on the left has two chips and the pile on the right has 1 chip .
- Two Players, I and II, take turn to remove chips.
- The player can choose one of the pile and then remove any number of chips from this pile.
- The player who removes the last chip loses.
- The winner wins $a from the loser.


I
L2
L1
R1
II
II
II
r1
l1
r1
l1
l2
I
I
I
R1
L1
L1
1
-1
-1
1
-1
1
1
-1
-1
1
1
-1

The Extensive Form of a Game

The extensive form, is built on the basic notions of position and move.

This is a most detailed description of a game. It tells exactly which player should move, what are the choices, the outcomes, the information of the players at every stage, and so on.

In the extensive form, games are sequential, interactive processes which moves from one position to another in response to the wills of the players or the whims of chance.

It will justify and give meaning to more abstract concepts such as strategy.

Three new concepts make their appearance in the extensive form of a game: the game tree, chance moves, and information sets.

The Game Tree. The extensive form of a game is modeled using a directed Graph.

We first define a directed graph.

A **directed graph** is a pair (T, F) where T is a nonempty set of vertices and F is a function that gives for each x ∈ T a subset, F(x) of T called the followers of x.

The vertices represent positions of the game.

The followers, F(x), of a position, x, are those positions that can be reached from x in one move.

A **path from a vertex** t_0 **to a vertex** t_1 is a sequence, x_0, $x_1, \ldots, x_n$, of vertices such that $x_0 = t_0$, $x_n = t_1$ and x_j is a follower of x_{j-1} for $i = 1, \ldots, n$. For the extensive form of a game, we deal with a particular type of directed graph called a tree.

The path from a vertex to one of its followers is called an edge and we may think of edges to represent moves from a given position.

A path from a vertex t_0 to a vertex t_1 then represents the whole history of play in the game moving from position t_0 to t_1.

Definition. A **tree** is a directed graph, (T,F) in which there is a special vertex, t_0, called the root or the initial vertex, such that for every other vertex $t \in T$, there is a UNIQUE path beginning at t_0 and ending at t.

The existence and uniqueness of the path implies that a tree is connected, has a unique initial vertex, and has no circuits or loops.

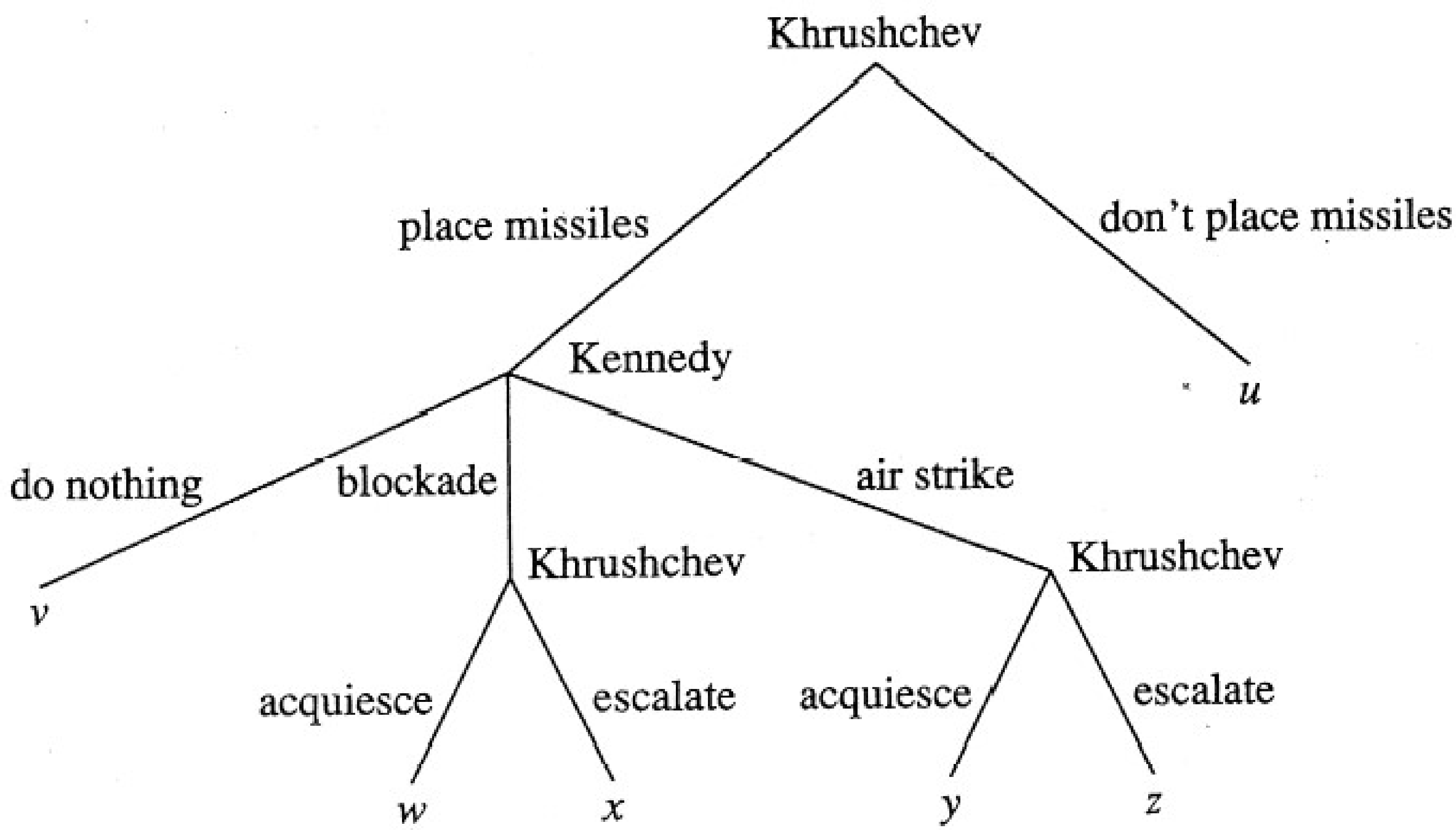
Therefore, vertices are in one-to-one correspondence with paths starting from the root, i.e. past moves leading to this position. Therefore, at each vertex we know the complete history of play moving from the root to this vertex.

The Kuhn tree of a game:

When a tree is used to define a game,

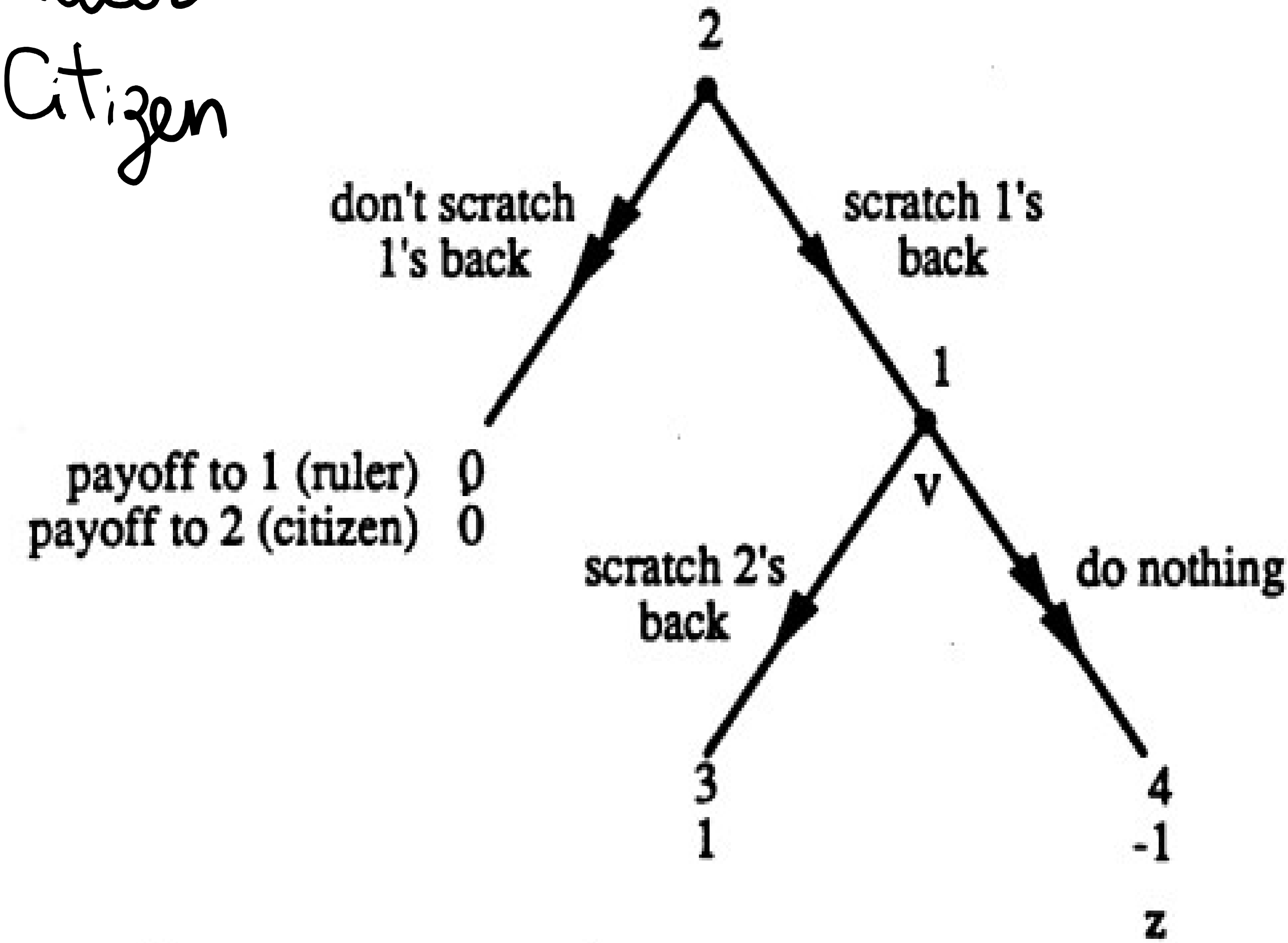
the vertices are interpreted as positions and the edges as moves.

Each nonterminal position is assigned either to the player responsible to choosing the next move or to chance.

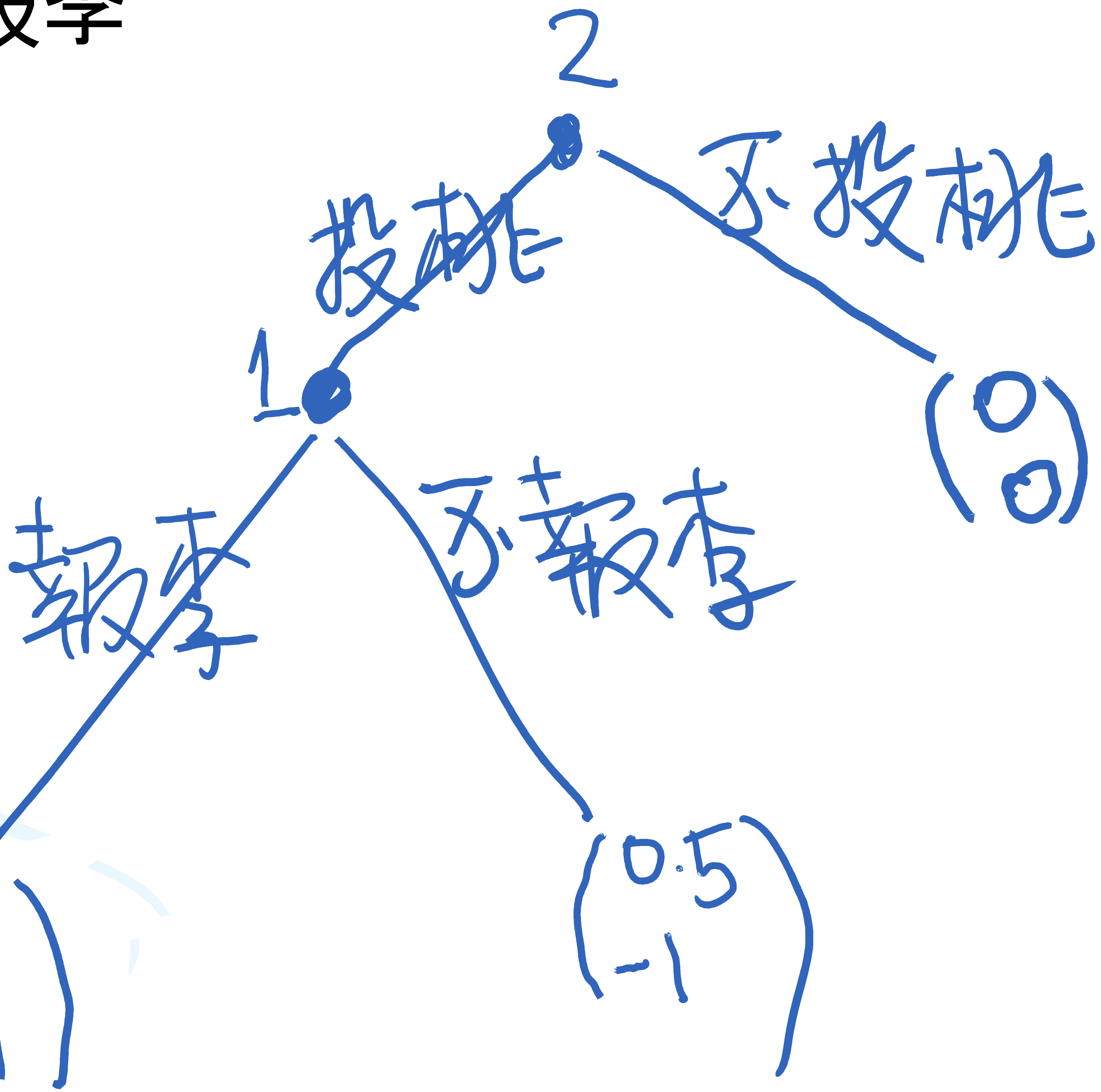

Khrushchev
place missiles
don't place missiles
Kennedy
u
do nothing
blockade
air strike
v
Khrushchev
Khrushchev
acquiesce
escalate
acquiesce
escalate
w
x
y
z

"If you scratch my back I'll scratch your back."

1 : Ruler
2 : Citizen

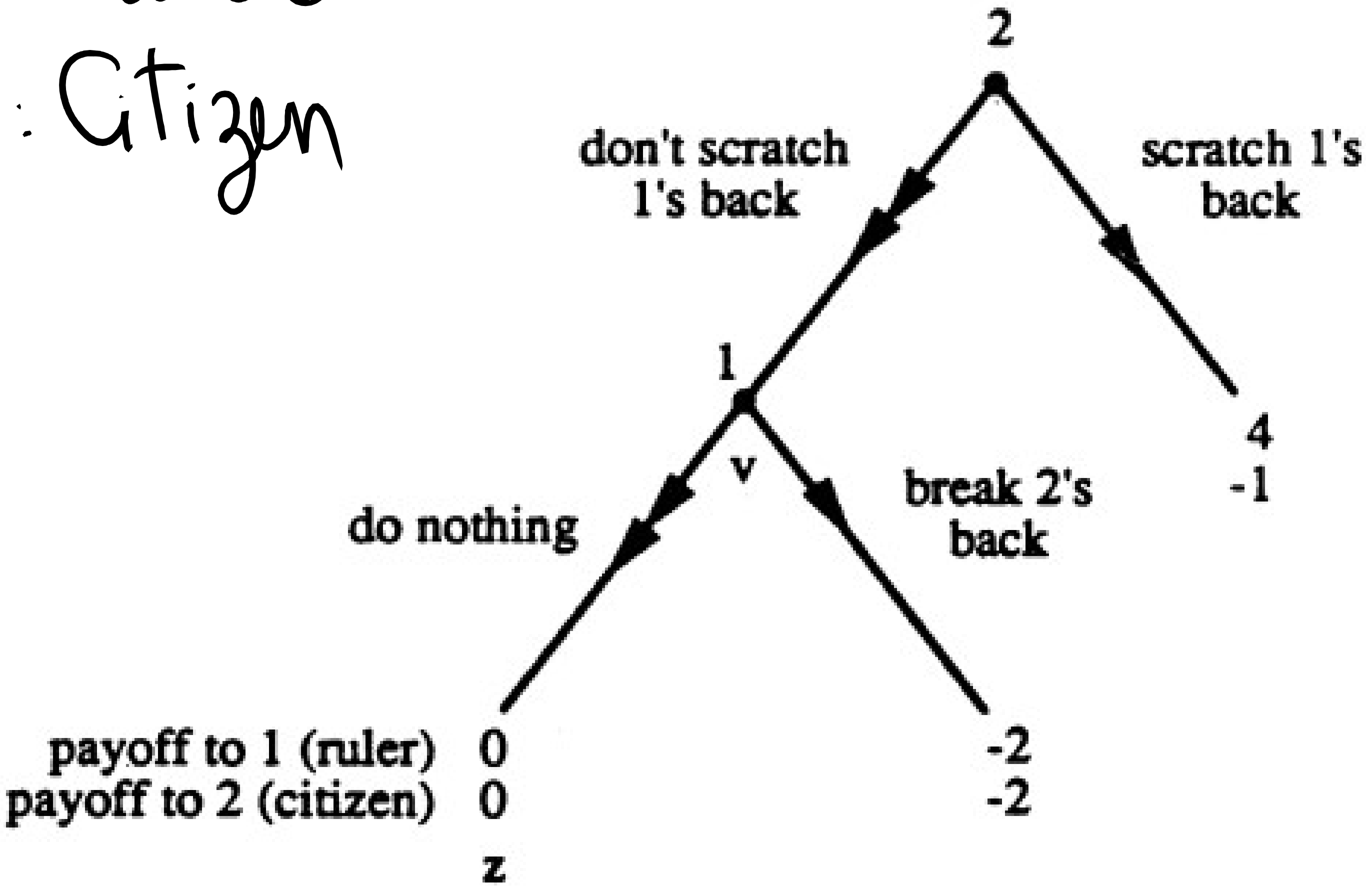


投桃报李

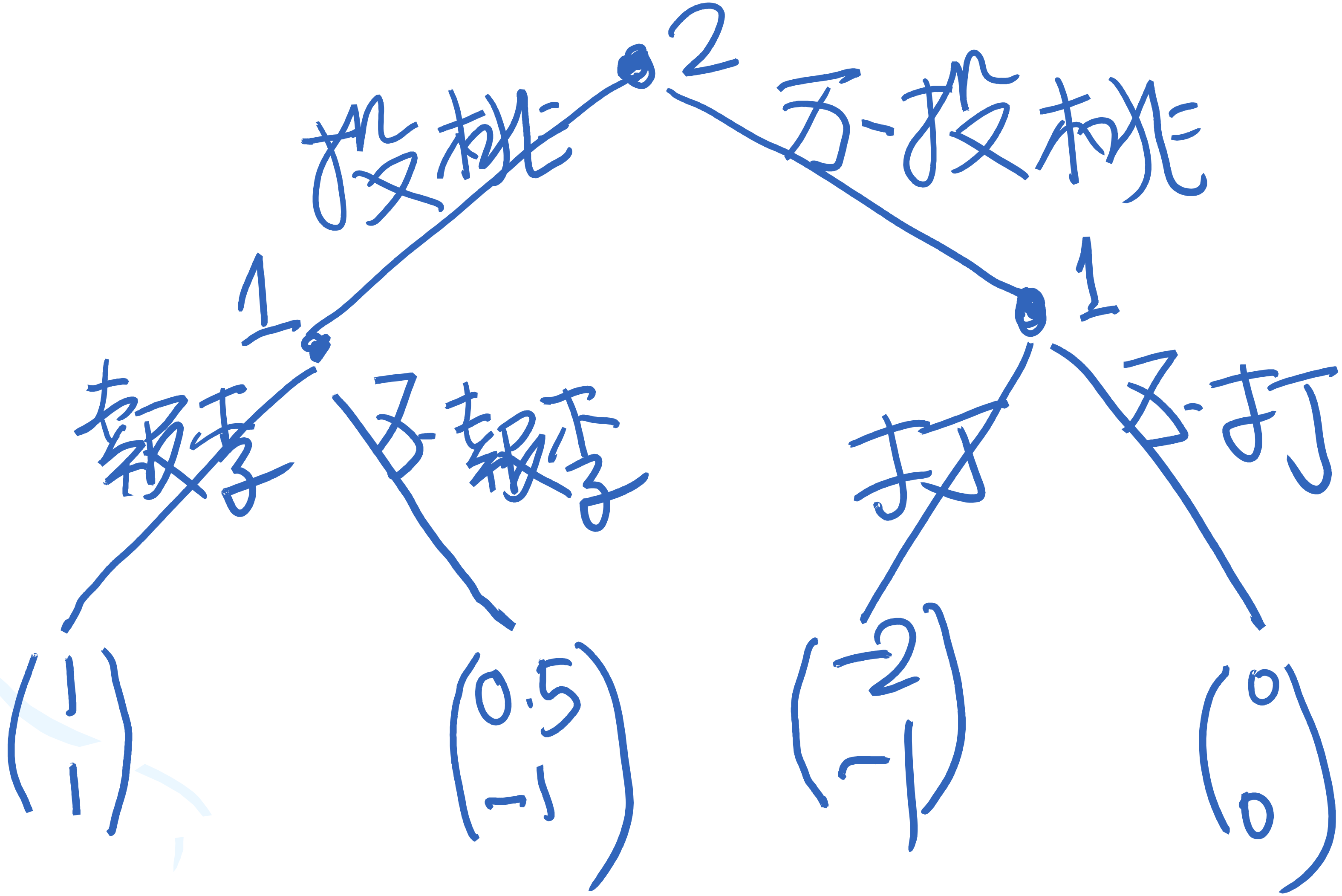


"Scratch my back or else I'll break your back."

1 : Ruler
2 : Citizen



投桃报李，不投桃就打你

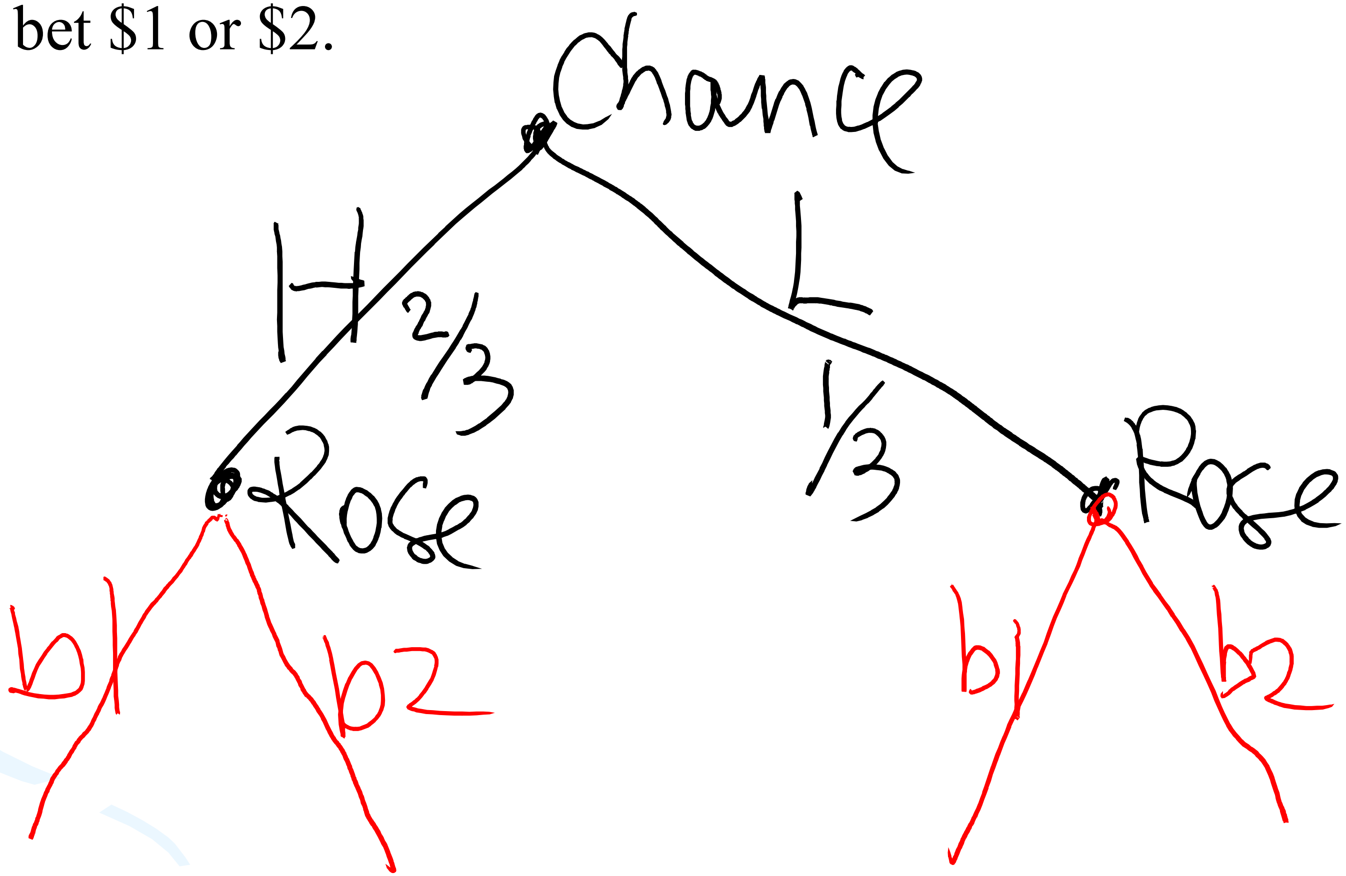


Chance Moves. Many games involve chance moves. Examples include the rolling of dice in board games like monopoly or backgammon or gambling games such as craps, the dealing of cards as in bridge or poker, the spinning of the wheel of fortune, or the drawing of balls out of a cage in lotto.

It is assumed that the players are aware of the probabilities of the various outcomes resulting from a chance move.

To incorporate chance moves in the Kuhn tree, we introduce a new player called Chance (Dame Fortune). The moves from Chance carry different probabilities. The probabilities assigned to the edges leading out of each chance vertex must be displayed.

Example: Two cards marked H (High) and one card marked L (Low) are placed in a hat. Rose draw a card and inspects it. She may then bet $1 or $2.



The root of the game starts the game and the terminal vertices represent various possible ending of a game.

Play starts at the initial vertex and continues along one of the paths eventually ending in one of the terminal vertices. Each directed path from the root to terminal vertex describes a possible course of play.

Each terminal vertex should carry some indication of the outcome of the game; it typically takes the form of a payoff vector, indicating how much each player wins or loses.

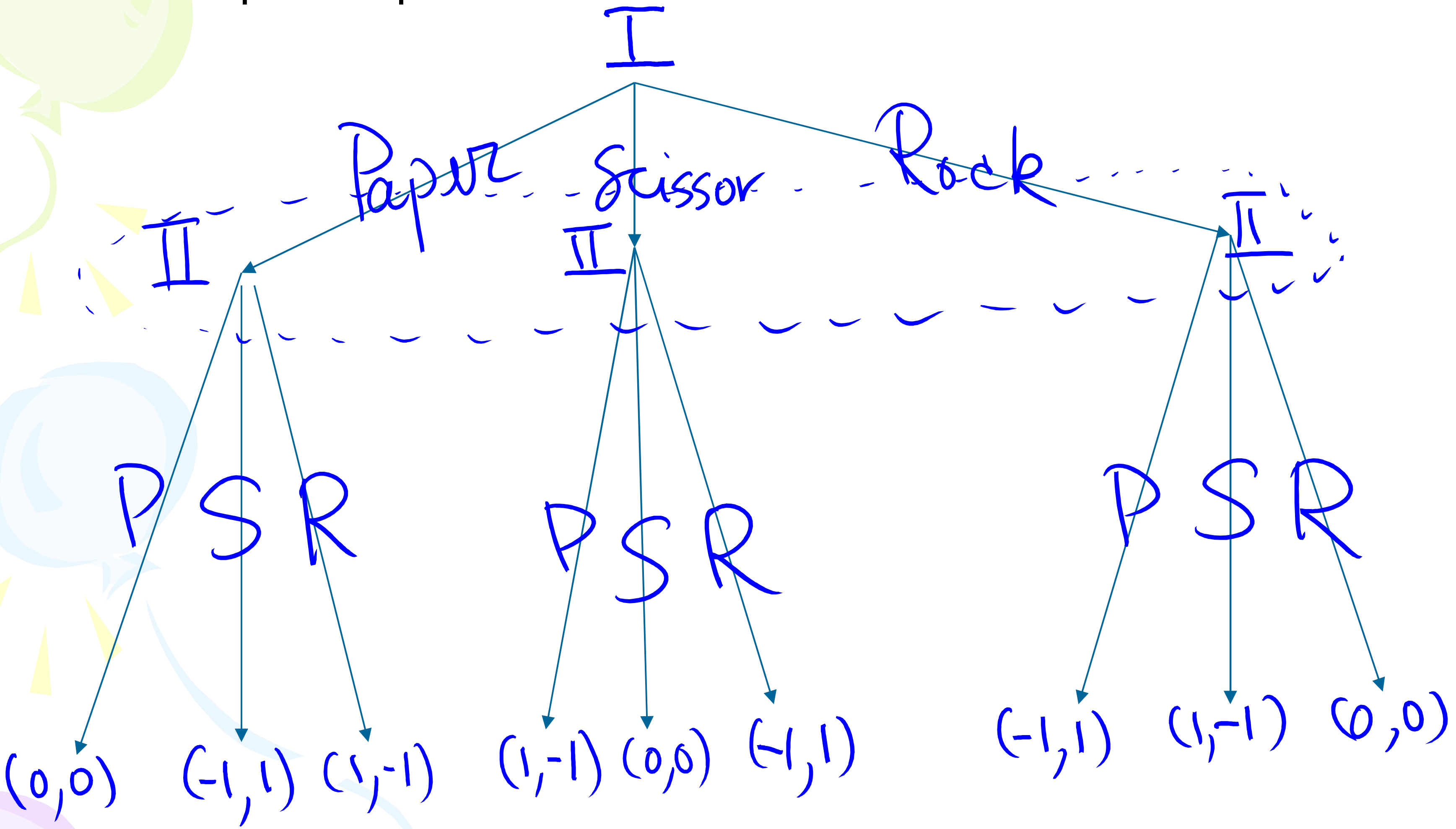
Information. Another important aspect we must consider in studying the extensive form of games is the amount of information available to the players about past moves of the game.

When a player choosing a move is uninformed about some of the previous moves (concealed moves, simultaneous moves etc.), we draw little "balloons" around set of vertices which the player making the choice cannot discriminate.

These are called information sets, and the player in question must make the same choice for all vertices in the set.

Edges issuing from vertices in the information set must be labeled the same.

Example: Paper-Scissor-Rock



Tian Ji Horse Race (田忌赛马):

General Tian Ji was a high official in the country Qi. He likes to play horse racing with the king and others.

Both of Tian and the king have three horses in different classes, namely, regular, plus, and super. Three races are to be held.

Sun Bin advised Tian Ji in using a rather simple strategy to win money from the king: using his regular class horse race against the super class from the king, his plus class horse race against the regular class from the king, his super class horse race against the plus class from the king.

What is the game tree? Is it the same as Paper-Scissor-Rock?

In summary, the following are the basic elements of a Kuhn tree.

1. A finite set N, representing the players.

2. A finite rooted tree representing positions and moves. The root represents the start of the game and the terminal vertices the possible conclusions.

3. Terminal vertices represent the possible conclusions of the game. The payoff to each player is labeled at this vertex as a vector.

4.Vertices are labeled to name of the player and Chance. A move by them at these positions means choosing a particular edge leading out from this vertex.

5. Edges coming out from Chance should be labeled with the probability of this particular occurrence.

6. Information sets: The vertices of each player are partitioned into information sets. The player labelled at this information set cannot distinguished the vertices within this information set.

Hence, there are same number of edges with the same label leading out from each vertex in the information set.

Example: Given the following Kuhn Tree,

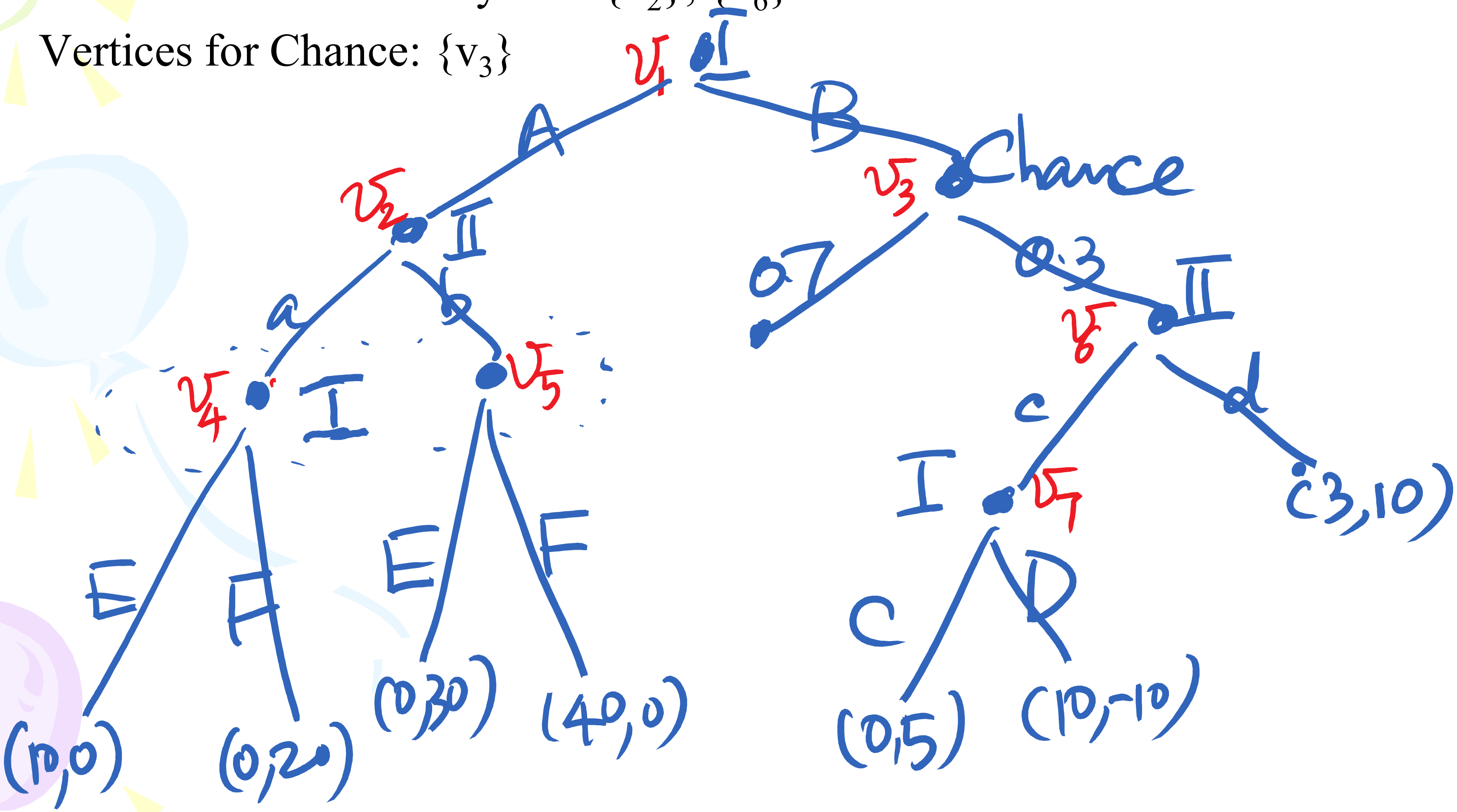
Vertices for Player I: $\{v_1, v_{4,}, v_5, v_7\}$

Information sets for Player I: $\{v_1\}, \{v_4, v_5\}, \{v_7\}$

Vertices for Player II: $\{v_2, v_6\}$

Information sets for Player II: $\{v_2\}, \{v_6\}$

Vertices for Chance: $\{v_3\}$



Games of Perfect Information. A game of perfect information is a game in extensive form in which each information set of every player contains a single vertex.

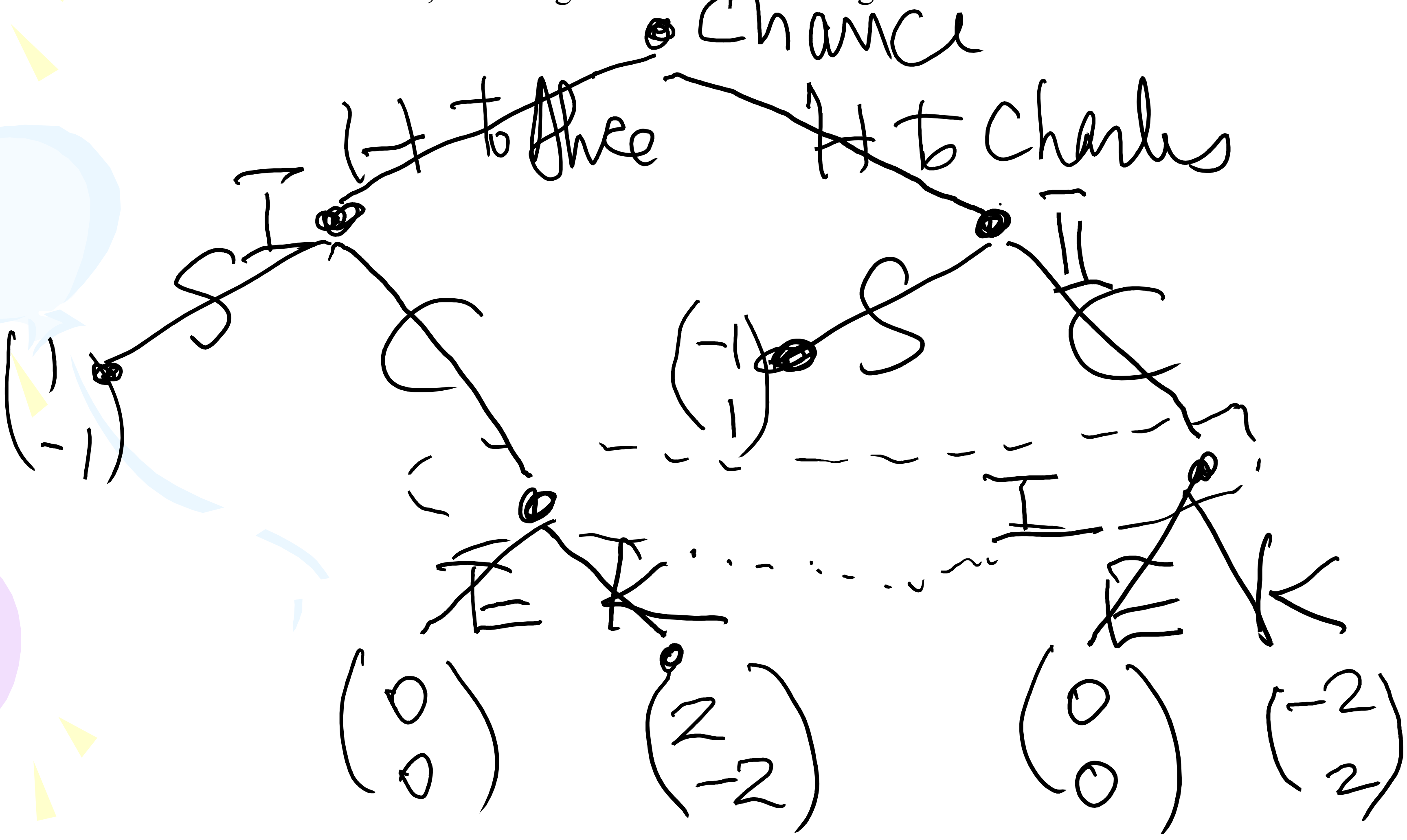
Games in which players remember all past information they once knew and all past moves they made are called games of **perfect recall**.

Games in which both players know the rules of the game, that is, in which both players know the Kuhn tree, are called **games of complete information**.

Games in which one or both of the players do not know some of the payoffs, or some of the probabilities of chance moves, or some of the information sets, or even whole branches of the tree, are called **games with incomplete information**.

Example (Kuhn 1953): Consider a 2-player game in which Player I consists of two agents, Alice and her husband Bill, and Player II is a single person, Charles. Two cards, one marked “High” and the other “Low”, are dealt at random to Alice and Charles. The person with the High card receives \$1 from the person with the Low card, and then has the choice of stopping or continuing the play. If the play continues, Bill, not knowing the outcome of the deal, instructs Alice and Charles either to exchange or to keep their cards. Again, the holder of the High card receives \$1 from the holder of the Low card, and the game ends. Draw the game tree.

Example (Kuhn 1953): Consider a 2-player game in which Player I consists of two agents, Alice and her husband Bill, and Player II is a single person, Charles. Two cards, one marked “High” and the other “Low”, are dealt at random to Alice and Charles. The person with the High card receives $1 from the person with the Low card, and then has the choice of stopping or continuing the play. It the play continues, Bill, not knowing the outcome of the deal, instructs Alice and Charles either to exchange or to keep their cards. Again, the holder of the High card receives $1 from the holder of the Low card, and the game ends. Draw the game tree.



Imperfect Recall

Memento（记忆碎片）: Example from movie suggested by Larry Blume

One of the most common assumptions made by game theorists is that players have perfect recall - they remember everything that happened in the game up to now. Larry Blume writes "Rubenstein and Piccione convinced us that imperfect recall might be pretty weird. The movie Memento shows how strategies in such a game might have to be implemented."

In the movie, the main character suffers from short-term memory loss, and adopts several unique strategies to remind him of what has so far transpired.