Game Theory: Science of Strategic Thinking

- GT is the **mathematical study** of interaction among independent, self-interested agents.
- **Self-interest**:
  - Each agent has its own interests and preferences (aims, goals);
  - Agents tend to have (partially) conflicting interests;
  - These interests are reflected in (numerical) utilities that are consistent with the preferences;

    $$\text{if option A is preferred to B, then } u(A) > u(B)$$

  - Agents act to maximise their utility
- **Coalitional/Cooperative** vs **non-coalitional/non-cooperative** GT

Ingredients of interesting games

- Players: You against one or more opponent(s)
  - Opponent: other agents, other version of yourself, nature, lady luck, etc.
- Rules determine which actions can be taken, and what the corresponding pay-offs or utilities are;
  - actions and pay-offs: exogenous variables
- Maximize your pay-off: Everyone wants to win!
- **Competition and collaboration:** individuals or teams (non-cooperative and cooperative GT)

Cooperative versus Non-Cooperative Games

- **Non-Cooperative**
  - Selfish individuals, only consider their own interest;
  - Do **not coordinate** their actions in groups
    - Emergence Coordination might happen as "accident" of selfish behaviour
  - Agreements need to be self-enforcing (no "contracts"!)

- **Cooperative:** Binding commitments (*"contracts"*) allow groups of players to coordinate their actions
  - Non-transferable utility: pay-off of each individual increases!
    E.g. Stable marriage problem;
  - Transferable utility: need to find a fair way to divide the additional value (e.g. money) generated by collaboration:
    E.g. Shapley value

Simultaneous vs. Sequential Games

- **Simultaneous games:** players make their moves simultaneously, i.e. without knowing what the other players will do!
  - Rock-paper-scissors
  - Sealed bid auctions
- **Sequential games:** Sequence of successive moves by players who can see each other's moves:
  - Chess
  - Card games
  - Open cry auctions

Encoding utilities of actions: Matrix form

Simultaneous game: **two players, finite number of actions**

	Player 2 chooses *Left*	Player 2 chooses *Right*
Player 1 chooses *Up*	**4, 3**	**−1, −1**
Player 1 chooses *Down*	**0, 0**	**3, 4**

Normal form or payoff matrix of a 2-player, 2-strategy game

Encoding utilities of actions: Extensive Form

Sequential game: **Decision tree**

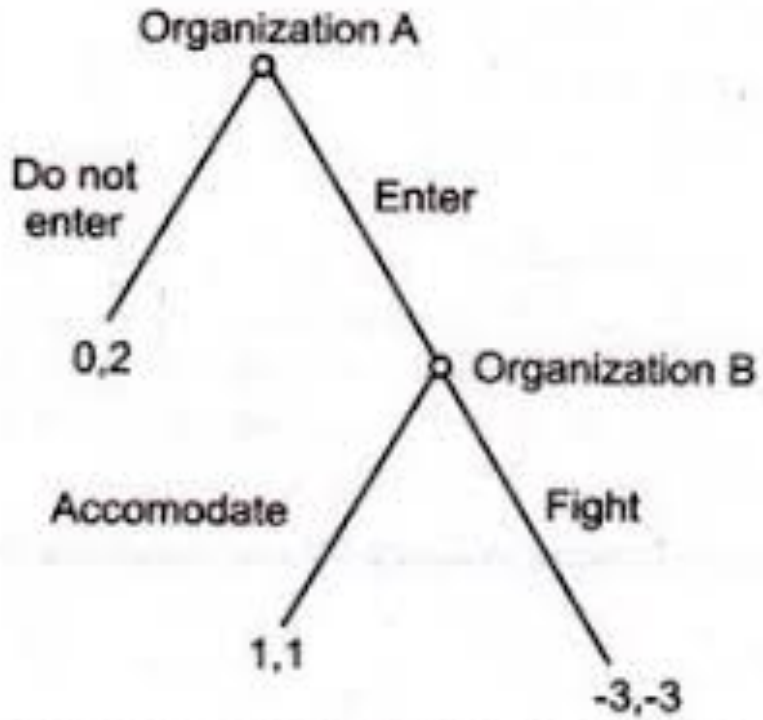


Figure-2: Extensive Form Games

Encoding utilities of actions: General case

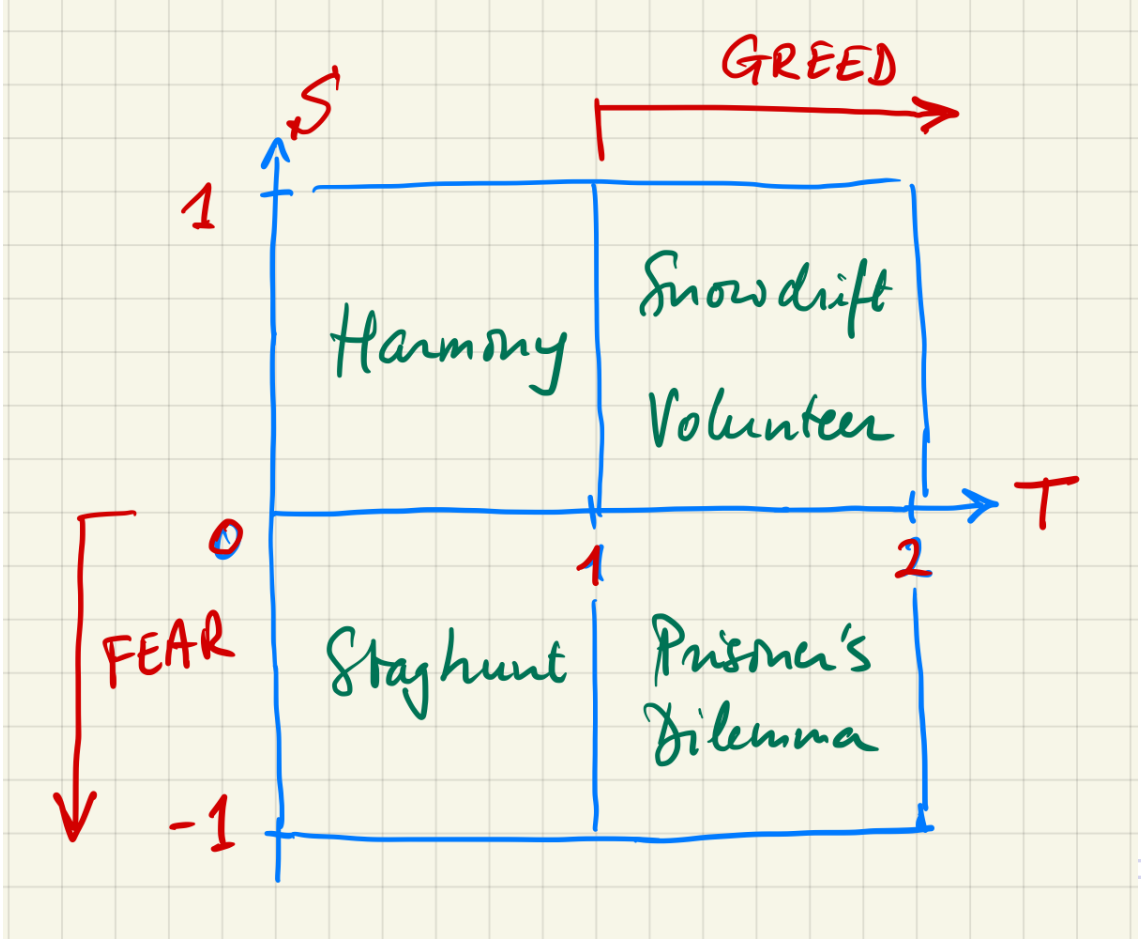
- Utility is a function of the joint action of players:
- Ultimatum game (one shot)
  - Player A can choose any fraction $0 \leq x \leq 1$ for himself, and offer the rest $(1 - x)$ to player B;
  - If player B accepts this offer, then that is the outcome. If he rejects it, then both get zero.

$$u_A(x) = \begin{cases} x \\ 0 \end{cases} \quad \text{and} \quad u_B(x) = \begin{cases} 1 - x & \text{if B accepts} \\ 0 & \text{if B rejects} \end{cases}$$

Important applicability issue: Rationality vs Behaviour (emotion)

2-parameter family of social dilemmas

	C	D
C	1, 1	S, T
D	T, S	0, 0



Overview of Topics in Game Theory Course

- Non-cooperative games
  - Matrix games (2 players, finite action sets)
  - Sequential games, e.g. bargaining
- Cooperative (coalitional) games;
  - Shapley value;
- Mechanism design (inverse game theory)
  - Vickrey auction

Game theory and strategic agents

- **Game theory** studies **multi-agent decision problems**, that is, problems in which **independent decision-makers interact**.
- What each agent does has an **effect on the other agents** in the group (through utility);
- **Assumptions:**
  - agents have **preferences** encoded in **utility function (pay-off)**
  - **self-interest:** agents strive to maximize their own pay-off;
  - **rational behaviour:** agents reason about the actions of other agents and decide rationally.

Normal-form Games (Matrix Games)

- **Players:**
  - make **simultaneous moves** and receive **immediate payoffs**;
  - **payoffs** are specified for the combinations of actions played.
- **Payoff matrix:**
  - Specifies for given action combination a $= (a_1, a_2, \ldots, a_n)$ the corresponding utility (pay-off) $u_i(\mathrm{a})$ for player $i = 1 \ldots n$

	Player 2 chooses *Left*	Player 2 chooses *Right*
Player 1 chooses *Up*	**4, 3**	**–1, –1**
Player 1 chooses *Down*	**0, 0**	**3, 4**

Normal form or payoff matrix of a 2-player, 2-strategy game

Formal definition of normal-form game

A n-person **normal-form game** is a tuple (N, A, u):

- N is a set of n **players (agents)**
- **Actions or Strategies** $A = A_1 \times A_2 \times \ldots \times A_n$ where each A_i is the set of actions available to agent i, i.e. set of allowable moves player i can make.
  An A-element a $= (a_1, a_2, \ldots, a_n)$ is called an **action profile**.
- **Pay-off or utility function:** u $: A \longrightarrow \mathbb{R}^n$ where u $= (u_1, u_2, \ldots, u_n)$ and each $u_i : A \longrightarrow \mathbb{R}$ is the corresponding utility function for player i. Notice, payoff u_i(a) for *each* agent depends on the *joint actions* of all agents.

Utility functions capture preferences

von Neumann and Morgenstern, 1944

If there exists a preference relation $\succcurlyeq$ on the outcomes of a game that satisfies a number of "natural conditions" (completeness, transitivity, substituability, decomposability, monotonicity and continuity), then there exists a function $u : \mathcal{O} \longrightarrow \mathbb{R}$ such that:

- $u(o_1) \geq u(o_2) \quad \text{iff} \quad o_1 \succcurlyeq o_2$
- $u(\{(o_1 : p_1), (o_2 : p_2), \ldots, (o_n : p_n)\}) = \sum_{i=1}^{n} p_i u(o_i)$

Strategies

- A player's **strategy** is the **algorithm** that determines the action the player will take at **any stage of the game.**
- **Pure strategy:** Select single action and play it.
- **Mixed strategy:** Select single action according to probability distribution and play it. :

Rationale? *matching pennies*

	Heads	Tails
Heads	1, -1	-1 , 1
Tails	-1, 1	1, -1

Mixed strategy: using randomness NOT to be *outsmart-ed* by opponent.

- **Strategy profile:** s $= (s_1, s_2, \ldots, s_n)$, i.e. one specified strategy for each agent.

Expected Utility for Mixed Strategies

- **Pure strategy:** (Expected) utility u_i for agent i selecting action a_i equals $u_i(a_i, a_{-i})$.
- **Mixed strategy:** Agent i plays strategy s_i which is a probability distribution over k possible actions:

$$s_i = \{(a_{i1}, p_{i1}), (a_{i2}, p_{i2}), \ldots, (a_{ik}, p_{ik})\} \qquad \text{(where } p_k = P(a_k)\text{)}$$

- **Expected utility** for mixed strategies:
  - agent i playing mixed strategy $s_i = \{(a_{i1}, p_{i1}) \ldots (a_{in}, p_{in})\}$
  - agent j playing mixed strategy $s_j = \{(a_{j1}, p_{j1}) \ldots (a_{jm}, p_{jm})\}$

$$EU_i(s_i, s_j) = \sum_{k=1}^{n} \sum_{\ell=1}^{m} u_i(a_{ik}, a_{j\ell}) p_{ik} p_{j\ell}$$

Analysing games: Solution concepts for games

Consider point of view of a **single (self-interested) agent:**

- Given all game information: **what strategy** should he adopt?
- Complicated: depends on **actions of other agents**!
- Solving a game means trying to predict its outcome.
- **From (weak) optimality . . .**
  - Pareto Optimality
  - Best Response (BR) given the actions of the other agents;
  - Iterated elimination of strictly dominated strategies (IESDS)
- **. . . over strategies with weak guarantees . . .**
  - Regret minimisation, Maximin and Minimax
- **. . . to Equilibrium,** i.e. no incentive to deviate:
  - Nash equilibrium (John Nash, 1950)

Pareto optimality

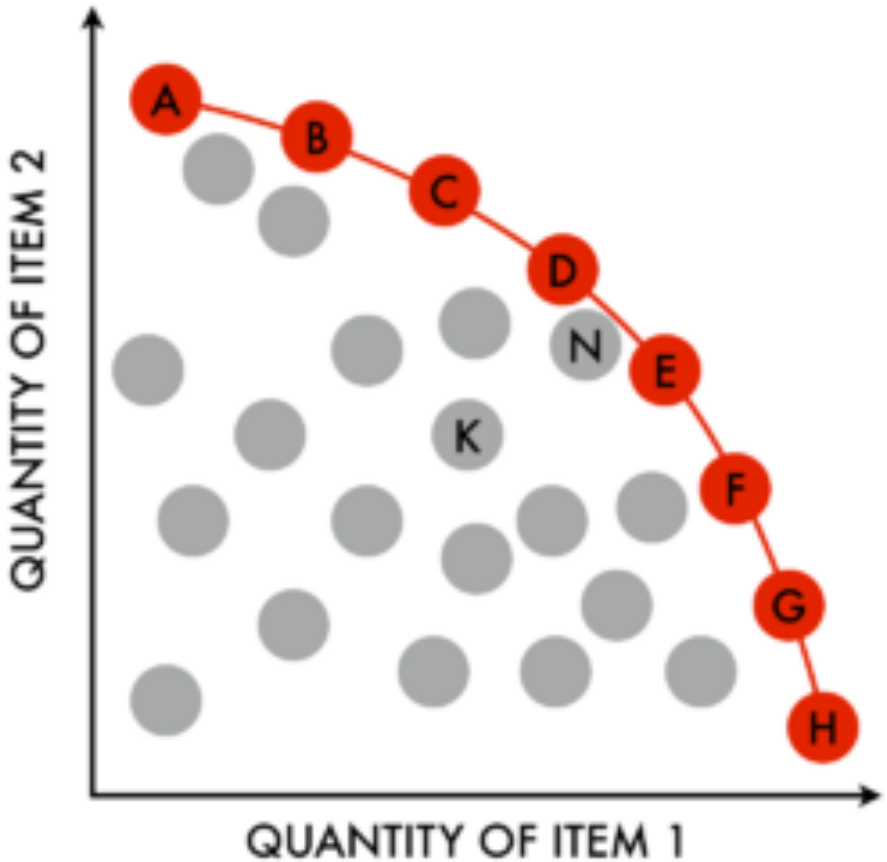
Pareto optimality is a **solution property** (not solution concept itself)

A joint action/strategy profile a is **Pareto dominated** by another joint action a′ if $u_i(\mathrm{a}') \geq u_i(\mathrm{a})$ for all agents i and $u_j(\mathrm{a}') > u_j(\mathrm{a})$ for some j.

A joint action/strategy profile a is **Pareto optimal** if there is no other joint action a′ that Pareto dominates it.

Pareto dominance defines a **partial ordering** over strategy profiles.

Pareto Front



Domination for strategies

Let s_i and s_i' be two strategies for player i, and S_{-i} set of all strategy profiles for the other players:

- s_i **strictly dominates** s_i' if

$$u_i(s_i, s_{-i}) > u_i(s_i', s_{-i}) \qquad \forall s_{-i} \in S_{-i}$$

- s_i **weakly dominates** s_i' if
  1. $u_i(s_i, s_{-i}) \geq u_i(s_i', s_{-i}) \qquad \forall s_{-i} \in S_{-i}$, and
  2. $u_i(s_i, s_j) > u_i(s_i', s_j)$ for at least one $s_j \in S_{-i}$

Dominant and dominated strategies

- **(Strictly/Weakly) Dominant Strategy:** (strictly/weakly) dominates every other strategy of the agent;
- **Strictly/Weakly) Dominanted Strategy:** is (strictly/weakly) dominated by at least one of this agent;

- A strictly dominated strategy will never be the best response to anything!
- For a dominating strategy, we don't have to worry what the opponents are going to do!
- Dominance plays important role in **mechanism design.**

What NOT to do? IESDS: Iterated elimination of strictly dominated strategies

IESDS (a.k.a. What NOT to do?) is based on the following assumptions:

- It is **common knowledge** that all agents are rational.
- Rational agents **never** play strictly dominated actions.
- Hence, strictly dominated actions can be **eliminated.**

	left	centre	right
up	$13, 3$	$1, 4$	$7, 3$
middle	$4, 1$	$3, 3$	$6, 2$
down	$-1, 9$	$2, 8$	$8, -1$

What would IESDS predict in this game?

Iterated elimination of strictly dominated strategies (2)

Centre **strictly dominates** *right***.** Row player knows that column player will never play the dominated action *right*. Hence he can eliminate that action and only needs to consider the simpler game:

	left	centre
up	13, 3	1, 4
middle	4, 1	3, 3
down	-1, 9	2, 8

For the row player, action *middle* strictly dominates *down*; hence eliminate! We are left with the simpler game where *centre* dominates *left*:

	left	centre
up	13, 3	1, 4
middle	4, 1	**3, 3**

Best response

- **Best response** from agent i's point of view:
- Let's assume that we know the strategies of all the other agents, i.e. s_{-i} is known;
- Agent i's best response s_i^* to strategy profile s_{-i}, is (a possibly mixed) strategy $s_i^* \in S_i$ such that

$$s_i^* = BR_i(s_{-i}) \quad \Leftrightarrow \quad \forall s_i \in S_i : \; u_i(s_i^*, s_{-i}) \geq u_i(s_i, s_{-i}).$$

- Of course, . . . in a realistic setting we don't know the strategies of the other agents!
- **Not** solution concept, but **essential ingredient** for Nash eq.!
- BR dynamics yields equilibrium in some cases.

Best response

- Best response **not necessarily unique**!
- When the best response includes two (or more) actions, then the agent must be **indifferent** among them!
- In fact: **any mixture** of these actions would also be a best response (mixed) strategy.
- Indeed,
  - If a_{i1} and a_{i2} are best both best response actions to s_{-i}, then $u_i(a_{i1}, s_{-i}) = u_i(a_{i2}, s_{-i}) =: u_i^*$.
  - Then, for any mixed strategy $s_i = \{(a_{i1}, p_1), (a_{i2}, p_2)\}$:

$$u_i(s_i, s_{-i}) = p_1 u_i(a_{i1}, s_{-i}) + p_2 u_i(a_{i2}, s_{-i}) = (p_1 + p_2) u_i^* \equiv u_i^*,$$

since $p_1 + p_2 = 1$.

Best Response Dynamics

- **Best Response Dynamics:**
  - Imagine the simultaneous game to be sequential;
  - Players take turns to play best response (BR) to opponent;
  - An equilibrium might be reached
- Example: in action profile (M, C) equilibrium is reached!

	L	C	R
U	$4, 3$	$2, 0$	$8, 2$
M	$8, 2$	$4, 6$	$-1, 1$
D	$6, -3$	$0, 0$	$1, -1$

- Counter-example: matching pennies produces a cycle!
- In finite game, converges to either equilibria or cycles!
  - This presages the concept of **Nash equilibrium**

Best response and IESDS

- Strictly dominated strategies are **never a best response;**
- **Church-Rosser property:** Order of elimination does not matter for IESDS (strict dominance)!
- Eliminating **weakly dominated** strategies might be too drastic!

Strategies with (weak) guarantees

- Focus on outcomes that can be guaranteed by your own actions
- **Reasons:**
  - Opponent-agnostic: Opponent's utilities might not be known
  - Following a different/hidden agenda:
    e.g. threatening or punishing opponent;
  - . . . others??
- **Strategies**
  1. Regret minimisation
  2. Safety (maximin) and Punishment (minimax) strategies

Regret minimisation

- **Regret** (for agent i) is the difference between the actual and maximal pay-off for a given action profile (s_i, s_j)

$$R_i(s_i, s_j) = u_i(BR_i(s_j)) - u_i(s_i, s_j) = \max_{s_i'} u_i(s_i', s_j) - u_i(s_i, s_j)$$

- For each action s_i , there is maximum regret depending on s_j:

$$R_i^{max}(s_i) = \max_{s_j} R_i(s_i, s_j)$$

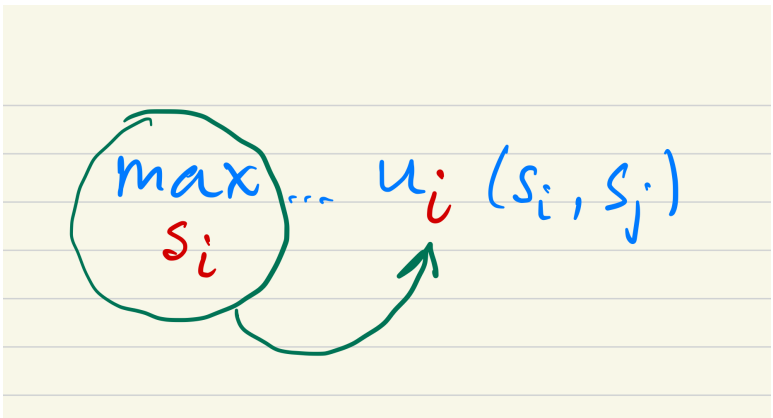
- **Regret minimisation** (minimax regret): agent i picks action s_i that mimimises max regret:

$$s_i^{rm} := \arg\min_{s_i} R_i^{max}(s_i) = \arg\min_{s_i} \max_{s_j} R_i(s_i, s_j).$$

- Actual solution concepts: allows an agent to choose a strategy with specific guarantees/properties;

Maximin Value and Safety Strategy

- Safety strategy: What is the best outcome i can secure for myself, no matter what the opponent does?
- No need to know the corresponding opponent's pay-offs!
- We consider player i's point of view: (i.e. maximise over s_i)



Maximin strategy (safety strategy): Algorithm

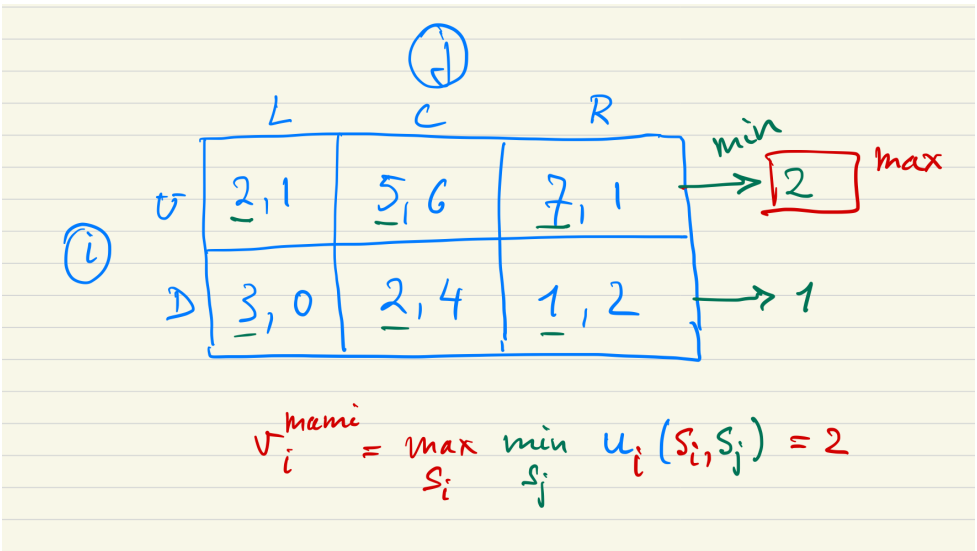
1. Ag_i computes for all his actions the worst possible outcome:

$$s_i \longrightarrow \min_{s_j} u_i(s_i, s_j)$$

2. Next, ag_i chooses action s_i to maximise his minimal pay-off:

$$v_i^{mami} := \max_{s_i} \min_{s_j} u_i(s_i, s_j)$$

Agent tries to maximise pay-off of worst possible outcome



Maximin Value and Strategy

Context: 2-player, general sum game

- Maximin **value** (or security level for WORST CASE) is the guaranteed minimal pay-off for agent i playing strategies in S_i:

$$v_i^{mami} := \max_{s_i \in S_i} \min_{s_j \in S_j} u_i(s_i, s_j)$$

- Maximin **strategy** (or safety strategy) for agent i maximizes his worst case pay-off (and therefore yields at least v^{mami} as utility):

$$s_i^{mami} = \arg \underbrace{\max_{s_i} \min_{s_j} u_i(s_i, s_j)}_{\text{security level}}$$

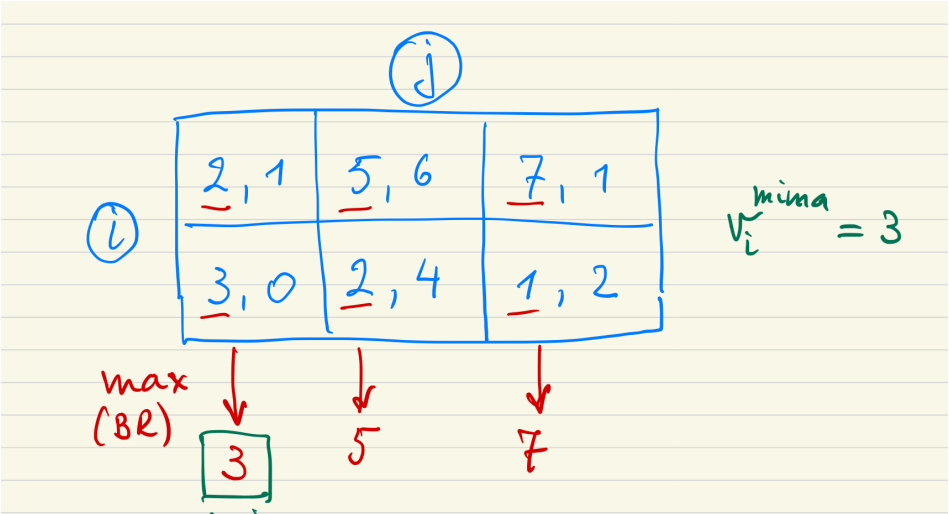
Maximin (safety) value and strategy

- **Why play maximin strategy?**
  - Pay-off guarantee! Highest pay-off agent i can guarantee for himself irrespective of the actions taken by other agent(s).
  - Worst case analysis: assume that opponent is malicious!

Punishment (or minimax) Strategy (of player j) yields Minimax Value (for player i)

1. Player i computes best response utility for each of j's strategies s_j: $BR_i(s_j) = \max_{s_i} u_i(s_i, s_j)$;
2. Then player j (who wants to punish i) picks action to minimise player i best pay-off. This punishment strategy of **player j** results in the minimax value for **player i**:

$$v_i^{mima} := \min_{s_j} \max_{s_i} u_i(s_i, s_j) = \min_{s_j} u_i(BR_i(s_j))$$



Algo: Minimax Value and Punishment Strategy

- **Minimax value** for agent i:
  - Given the strategy s_j of his opponent, agent i will play its **best response**, resulting in a pay-off:

$$\max_{s_i} u_i(s_i, s_j)$$

  - The opponent is aware of this and wants to *"punish" i* by *minimizing* this pay-off, yielding the minimax value for player i:

$$v_i^{mima} := \min_{s_j} \max_{s_i} u_i(s_i, s_j)$$

- The corresponding minimising strategy s_j^{mima} is called the minimax strategy for player j.
- If j plays his minimax strategy s_j^{mima}, then i cannot do better than v_i^{mima} (even if i plays best response $BR_i(s_j^{mima})$.

Comparing minimax and maximin value for player i

- $\forall s_i, s_j$:

$$\underbrace{\min_{s_j} u_i(s_i, s_j)}_{\phi(s_i)} \leq u_i(s_i, s_j) \leq \underbrace{\max_{s_i} u_i(s_i, s_j)}_{\psi(s_j)}$$

- Since the above inequality holds for all s_i and s_j, it follows:

$$\max_{s_i} \phi(s_i) \leq \min_{s_j} \psi(s_j)$$

- Hence:

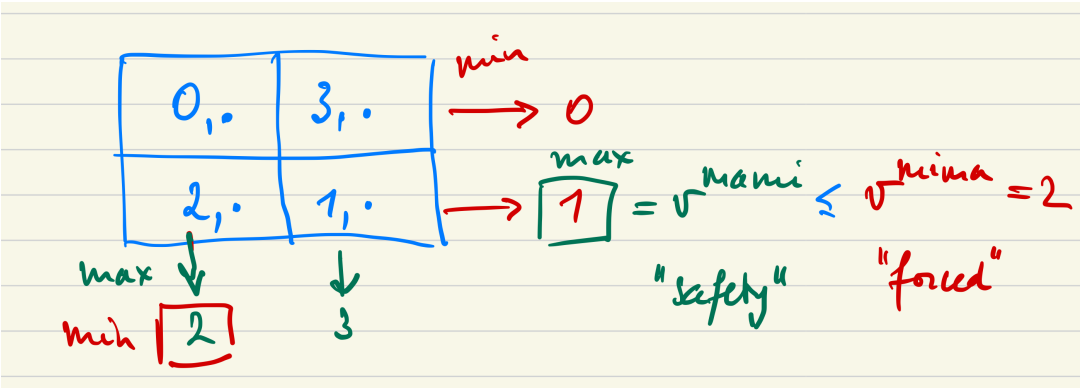
$$v_i^{mami} = \max_{s_i} \min_{s_j} u_i(s_i, s_j) \quad \leq \quad \min_{s_j} \max_{s_i} u_i(s_i, s_j) = v_i^{mima}$$

Minimax and Maximin Value for Player i

- In general:

$$v_i^{mami} = \max_{s_i} \min_{s_j} u_i(s_i, s_j) \quad \leq \quad \min_{s_j} \max_{s_i} u_i(s_i, s_j) = v_i^{mima}$$

- Your guaranteed pay-off is a is lower bound for the worst your opponent can force onto you!
- The worst your opponent can force onto you is an upper bound on your guaranteed pay-off.



Recap Safety and Punishment Strategy

- **Player *i*'s maximin strategy is safety strategy**
  - player *i* concerned about his own safety
    - Malicious or adversarial opponent
    - Multi-agent setting
  - strategy yields highest guaranteed outcome for player *i*
  - Viable solution algorithm.

- **Player *i*'s minimax strategy is punishment strategy**
  - *i*'s strategy is directed against player *j*
  - player *i* tries to minimize best (i.e. maximum) pay-off for *j*
  - Useful as threat (e.g. in repeated games);
  - *i*'s maximin **strategy** gives rise to *j*'s maximin **value**

Which is better? Regret Minimisation or Safety?

- Comparison for $(0 \leq T \leq 2, -1 \leq S \leq 1)$ parametrisation for social dilemmas.

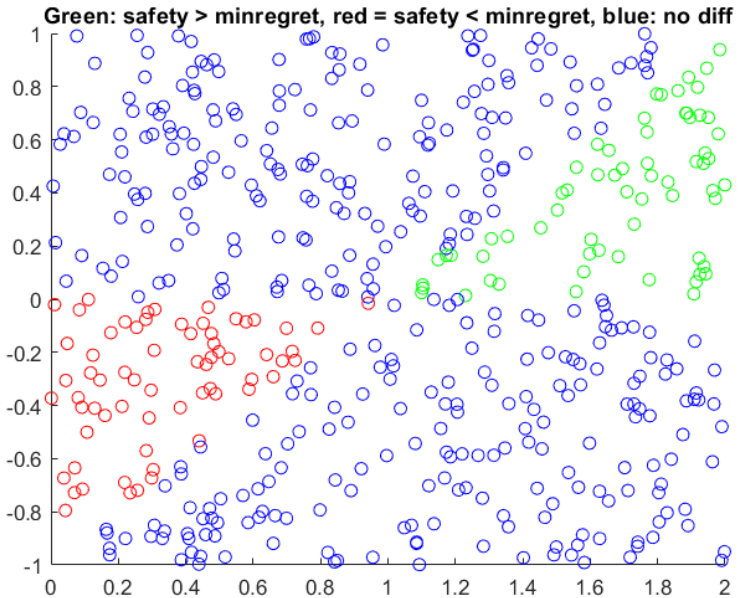


Figure: Utilities (assuming both agents play same strategy): : green: safety $>$ min regret, red $=$ min regret $>$ safety, blue: no difference

Nash equilibrium: Context

- Von Neumann's minimax theorem established game theory as a discipline;
- Nash equilibrium: Extension of von Neumann's minimax theorem
  1. From two person to *n* person game;
  2. from zero sum to general utilities
- For more info on the minimax theorem: See addendum;

Nash equilibrium (1)

- A **Nash equilibrium** (NE, 1950) is a solution concept based on *conditions* instead of an *algorithm*.
- **Mutual best response:** NE is joint strategy profile s^* such that **for each agent** i the strategy s_i^* is a **best response** to s_{-i}^*;
- **Formally:** A strategy profile $s^* = (s_1^*, s_2^*, \ldots, s_n^*)$ is a **strict NE** if:

$$\forall \text{agents } i, \forall s_i' \neq s_i^* : u(s_i^*, s_{-i}^*) > u(s_i', s_{-i}^*).$$

- **Strict** ($>$) versus **weak** ($\geq$) NE
- **No Regret/Self-enforcing:** a (strict) NE is a stable strategy profile for which no agent has an incentive to **unilaterally deviate**;

Nash's Theorem

Existence of Nash Equilibrium (Nash, 1950)

A **finite strategic game** (i.e. finite number of players and actions) always has **at least one Nash equilibrium** (allowing mixed strategies).

- A **pure** Nash equilibrium can be strict or weak;
- A **mixed** Nash equilibrium is necessarily weak;

Nash equilibrium: Nash's Theorem

- A **finite strategic game** is a game with a finite number of agents and a finite number of actions;
- A game may have **zero**, **one**, or **more pure-strategy NE**.
- If there's a **single** NE: natural solution concept, but might be (Pareto) sub-optimal!
- If there are **multiple NEs**: there might be no compelling reasons to pick a particular one; but ...
  - Utility dominant NE, Schelling's focal points
- Since humans are not always rational, Nash equilibria might not agree with experiments or observations..

Note on **multiple** Nash Equilibria

- A unique best alternative is exceptional.
- In general, non-cooperative game theory is plagued by a multiplicity of equilibria.
- Hence, prescriptions of how to act without any coordination or cooperation are in general impossible.
- Non-cooperative game theory tells us what to exclude from choice.
- Since in non-coop GT, there are no binding contracts, a player's announcements and promises in a pre-play phase are credible only, if they are totally in line with his best interests.

Computation of NE

- **Pure NE** **for each agent** i the strategy s_i^* is a **best response** to s^*_{-i} (mutual best response);
  - Matrix games (discrete state/action) space;
  - Continuous action space
- **Mixed NE**: make opponent indifferent (matrix games only);

Matching pennies: Mixed Nash Equilibrium

Nash equilibrium characteristics:

- At the intersection point $(p = 1/2, q = 1/2)$, players are simultaneously playing best response to each other;
- No player can do strictly better by unilaterally deviating:
  - If player 2 keeps playing $q = 1/2$ then player 1's utility $u_1(p, q = 1/2)$ can be computed as follows:

$$\begin{aligned} u_1(p, 1/2) &= (1 \cdot p + (-1) \cdot (1 - p) + (-1) \cdot p + 1 \cdot (1 - p)) \cdot \frac{1}{2} \\ &= 0 \qquad \textit{(independent of } p\textit{)} \end{aligned}$$

  Player 1:no incentive to change his strategy (i.e. change p)

  - Same consideration for player 2.

Computing mixed Nash equilibrium

Player 1: T or B, player 2: L or R

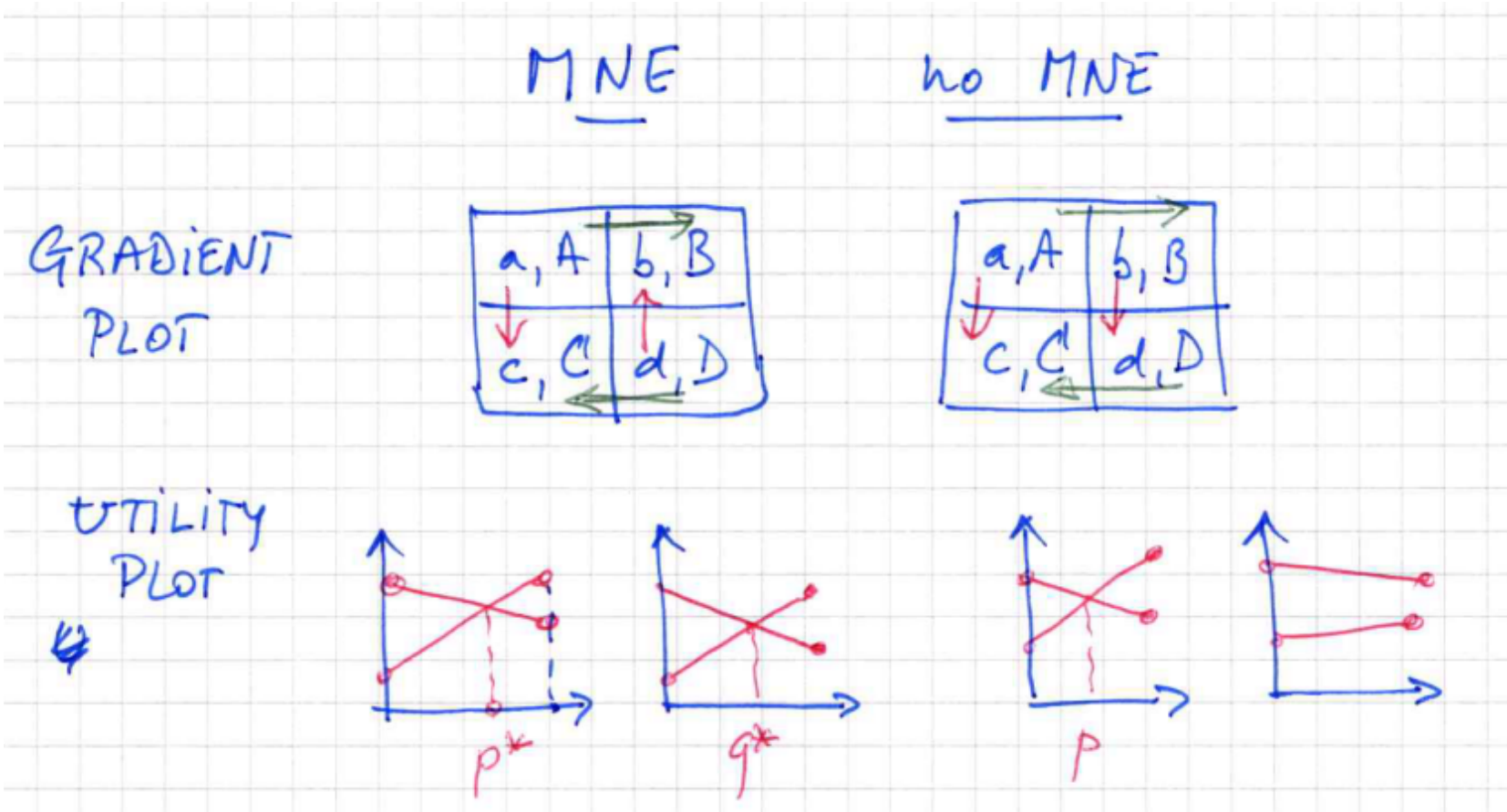
- **Strategies:** Player 1: T or B, Player 2: L or R
- Player 1: fix $p = p^*$, such that player 2 is indifferent (between L and R):

$$EU_2(p^*, L) = EU_2(p^*, R)$$

- Player 2: fix $q = q^*$, such that player 1 is indifferent (between T and B):

$$EU_1(T, q^*) = EU_1(B.q^*)$$

Some useful graphical representations



Fictitious Play (FP)

- FP uses simulation of many game iterations to learn about equilibria;
- FP refers to a dynamic process where at each stage, players play a (pure) best response to the empirical distribution of their opponent's play (which they interpret as a mixed strategy);
- **If FP converges (in distribution)**, the limit distribution coincides with mixed Nash strategies.
- Reminiscent of best response dynamics, but more general (includes not just last action, but whole history)
- **Form of learning**, works even opponent's utilities are unknown;

Vickrey auction: Second Price Auction

- **Vickrey Auction:**
  - n sealed bid auction for single item;
  - highest bid wins, but pays 2nd highest price;
- **Truth-telling** is (weakly) **dominant strategy;**
- NE: Neither winner nor loser(s) have incentive to deviate:
- **Winner:**
  - Higher: still winner, same price;
  - Lower: might lose, but if still winner, still paying 2nd price;
- **Loser:**
  - Higher: possibly winner, but at higher price;
  - Lower: still loser;
- Example of **mechanism design**.

Hawk or Dove

- Equilibria as a function of **exogenous pay-off variables**;
- **Exogenous variables** are imposed on the game (not by players);
- **Strategy: Hawk or dove:**
  - Two parties are in conflict over some good of value $2v > 0$;
  - Two doves share, each gets v;
  - Hawks fight, on average each gets half, but at a cost c (e.g. due to injury)
  - A dove is no match for a hawk and yields;
  - **Pay-off matrix:**

	Hawk	*Dove*
Hawk	$v-c,\ v-c$	$2v, 0$
Dove	$0, 2v$	$v,\ v$

- Different outcomes depending on cost of aggression c!

Hawk or Dove

- **Aggression is cheap** $(c < v)$ H is dominant, hence: NE = (H,H) with utility $(v - c, v - c)$;
- **Aggression is risky** $(c > v)$
  - Two PNE = (H,D) and (D,H)
  - One MNE at $P(H) = p^* = v/c$.
  - If risk/cost increases $c \uparrow$, then $p^* \downarrow$.

Strategic Effects

- Kicker has weak left side, so is inclined to kick less with left;
- Goalkeeper knows this, so anticipates less kicks to the left, so will tend to jump less to the left . . .
- Since kicker knows this, he might reconsider and kick more with left, since goalie has tendency to jump to the right . . . ,
- but goalie knows this too, so might reconsider . . .
- and so on . . .
- Is there a way out of infinite regress???

Strategic Effects

- Mixed Nash equilibrium at:

$$p^* = \frac{1}{1+a} = 1 - q^* \quad \Rightarrow \quad p^* > 1/2, \quad q^* < 1/2$$

- Zero-sum game: Utilities for kicker and goalie:

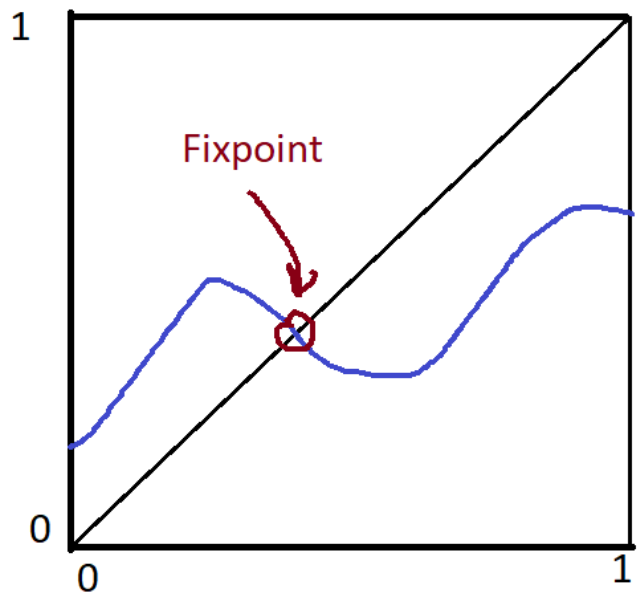
$$EU_1(p^*, q^*) = \frac{a}{1+a} = -EU_2(p^*, q^*)$$

- Notice that $p^* \to 1$ as $a \to 0$, i.e. if left kick is powerless ($a \approx 0$), make sure to kick left ($p^* \approx 1$). Is NE the right tool to think about this??

Aside: Sperner's Lemma (1928) and fix-point theorems

Defintion: Fix-point

Point $a \in A$ is a **fix-point** for function $f : A \longrightarrow A$, iff $f(a) = a$.



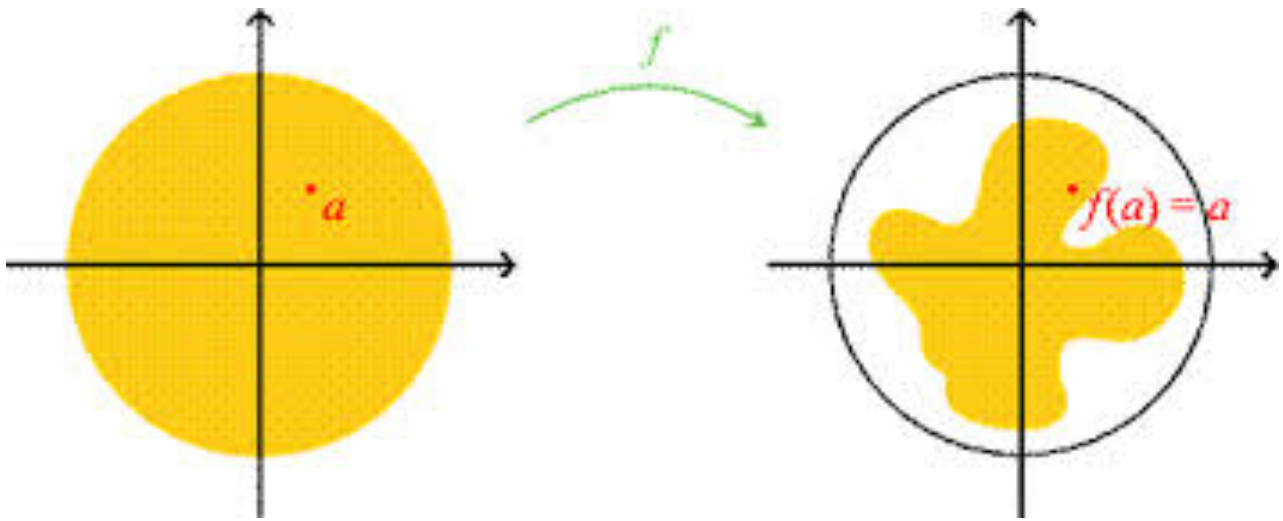
f: [0,1] ---> [0,1] continuous

Sometimes(!), fixpoints can be computed using **function iteration.**

Nash Theorem is based on Fixed-Point Theorem

Brouwer's Fixed Point Thm

Let $K \subset \mathbb{R}^n$ be a compact and convex, and $f : K \longrightarrow K$ continuous. Then f has a fix-point in K, i.e. there exists a $x_0 \in K : f(x_0) = x_0$.



: **Non-constructive existence proof!**

Nash equilibrium: Computational aspects

- Finding a Nash equilibrium for 2-player zero-sum games can be done efficiently by formulating a linear program. Notice that in this case: NE = minimax = maximin.
- Finding a Nash equilibrium is not known to be NP-complete because it is not a decision problem
- PPAD (polynomial parity argument, directed version) is a class describing problems for which a solution always exists
- Daskalakis, Goldberg, and Papadimitriou showed that finding a sample Nash equilibrium of a general-sum finite game with two or more players is PPAD-complete (i.e. "difficult!").

Summary

- Game theory studies utility-based multiagent decision making.
- Solving a game means trying to predict its outcome.
- Rational agents never play strictly dominated actions.
- No agent has an incentive to **unilaterally deviate** from a **Nash equilibrium**.
- In finite games, there's always a NE (possibly in mixed strategies).
- Nash equilibria need not be Pareto optimal.

Sequential games

- **Normal-form games:**
  - Simultaneous moves by players
  - Central solution concept: Nash equilibrium
- **Sequential games:**
  - Players move in succession, observe (at least partially) prior moves by opponents
  - **Perfect versus imperfect information:** what exactly is known about previous moves?
    - players have full knowledge of all the preceding moves (perfect information)
    - players might not know the complete game history till then (imperfect information);
  - **Model for many sequential interactions** in games, politics, economics, etc

Simultaneous vs. Sequential Games

- **Simultaneous games:** players make their moves simultaneously, i.e. without knowing what the other players will do!
  - Rock-paper-scissors
  - Sealed bid auctions
  - Cournot's duopoly model
- **Sequential games:** Sequence of successive moves by players who can see each other's moves (to some extent – see next slide):
  - Chess
  - Card games
  - Open cry auctions
  - Stackelberg's duopoly model
  - Negotiation (Rubinstein's model)

Extensive form representation of sequential game

- Visualisation of temporal relationships (game tree)
- **Extensive form** is finite game representation that **does not assume** that players act **simultaneously**;
- Can be converted in normal form representation (*possibly exponentially larger!*)
- **Game tree:** makes **temporal structure** explicit

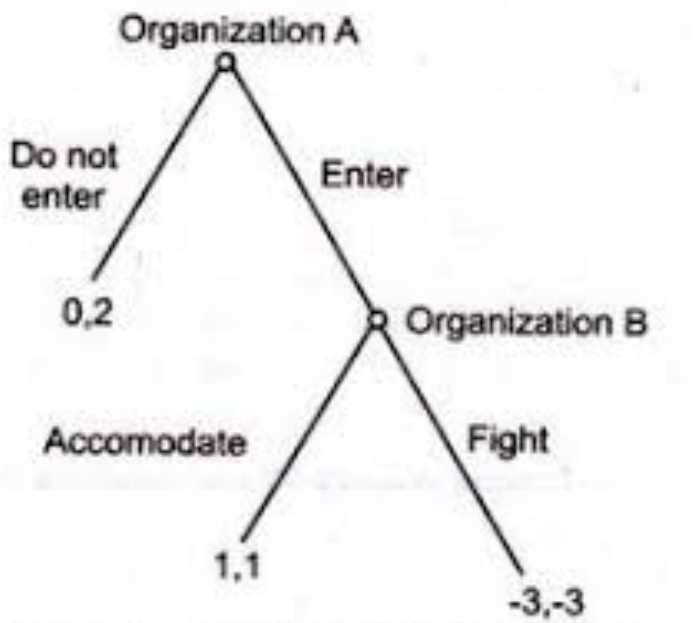


Figure-2: Extensive Form Games

Information in Game Theory

	PERFECT complete history known to all players	**IMPERFECT** unaware of actions taken by others
COMPLETE NO private info agents, actions, payoffs known	E.g. chess	Simultaneous games Information sets
INCOMPLETE private info e.g. private valuation	• Open cry auction • Different types of opponents	Sealed bid auction

Aside: Types of knowledge

- **Mutual knowledge** is
  - known to all players,
  - but players do not know that others know
  - e.g. *the elephant in the room* , solutions to homeworks
- **Common knowledge** is
  - known to all players,
  - and all players know all others know ...
  - and all players know all others know that all others know ...
  - and so on ...
  - e.g. In continental Europe one drives on the RHS of the road

Major Ideas in Non-Cooperative Game Theory

	SIMULTANEOUS (STATIC)	**SEQUENTIAL** (DYNAMIC)
Complete information NO private info	Nash equilibrium	Backwards induction, Subgame-perfect NE: *discard NE based on non-credible threats*
Incomplete information private info	Bayesian Nash eq.	Perfect Nash eq.

Sequential Games with Perfect Information

Solution strategy: Backward induction

Protoype: two-player, sequential-move game:

- Player 1 chooses action $a_{11} \in A_1$;
- Player 2 observes a_{11} and then chooses action $a_{21} \in A_2$;
- ...
- Hereafter, both players receive pay-off: $u_1(a_{1n}, a_{2n})$ and $u_2(a_{1n}, a_{2n})$ respectively;

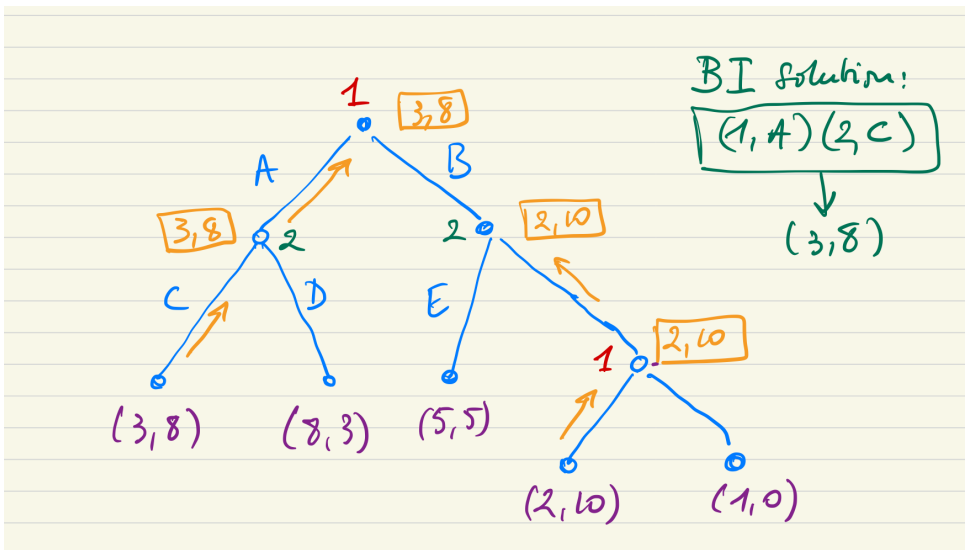
Examples:

- Various board and card games (e.g. chess, go, etc)
- Stackelberg's sequential-move version of Cournot's duopoly;
- Rubinstein's bargaining model

Solving Extensive Form Games using Backward Induction

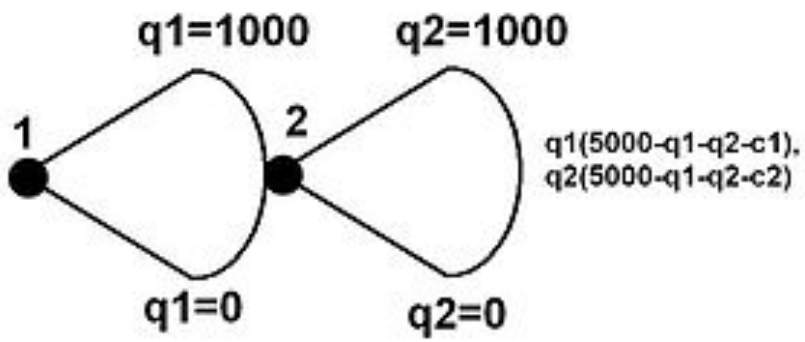
Backward induction

- Basic assumption:
  - Players believe that all future play will be rational
  - and condition decisions on what they expect in future;
- Algorithm
  - Start at leaf-nodes: easy decision as only one player involved;
  - Propagate decisions and utilities to root;
  - The solution path (A, C) is called the equilibrium path



Sequential game with continuous actions
Stackelberg duopoly

- Sequential version of Cournout duopoly
- In a **Stackelberg game**,
  - one player (leader) moves first,
  - and all other players (followers) move after him.
  - Continuous action space but can also be solved using backward induction



Stackelberg Duopoly

- Two firms produce a bland product (e.g. bottled water, airline seats).
  - Bland product: customers don't care which firm they buy from.
- Firm 1 moves first and decides to produce a total quantity q_1.
- Firm 2 observes this move, then decides to produce q_2.
- The market price (per unit) decreases (linearly) as the total amount produced $(q_1 + q_2)$ increases:

$$P(q_1, q_2) = \alpha - \beta(q_1 + q_2) \qquad (\alpha, \beta > 0)$$

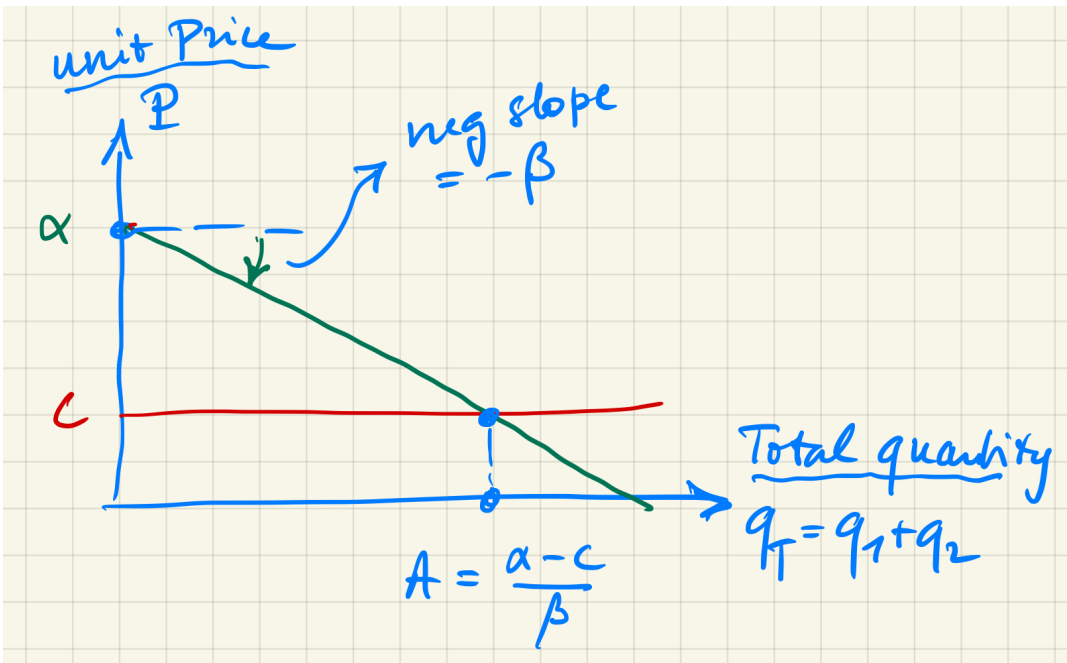
- Assume that both firms can produce the product at a fixed unit cost c. Hence the pay-off for each firm equals:

$$u_i(q_1, q_2) = (P(q_1, q_2) - c)q_i$$

Stackelberg Duopoly

$$
\begin{aligned}
u_i(q_1, q_2) &= (P(q_1, q_2) - c)q_i \\
&= (\alpha - \beta(q_1 + q_2) - c)q_i \\
&= \beta\left(\frac{\alpha - c}{\beta} - (q_1 + q_2)\right) q_i \\
&= \beta\left(A - (q_1 + q_2)\right) q_i \qquad \text{where } A = (\alpha - c)/\beta.
\end{aligned}
$$

A is the total quantity at which price equals cost ($P = c$), hence a quantity in excess of A is not economically viable.



Solving Stackelberg competition using backward induction

- For an observed quantity q_1 we compute $q_2^* = BR_2(q_1)$:

$$\frac{\partial u_2}{\partial q_2} = \frac{\partial}{\partial q_2}\beta\left(A - (q_1 + q_2)\right) q_2 = \beta(A - (q_1 + 2q_2))$$

- Hence:

$$\frac{\partial u_2}{\partial q_2} = 0 \quad \Longrightarrow \quad q_2^* = \frac{1}{2}(A - q_1)$$

Solving Stackelberg competition using backward induction

- **Given the anticipated optimal response q_2^* of firm 2**, what is best action q_1^* for firm 1?
- Optimal utility:

$$\begin{aligned} u_1(q_1, q_2^*) &= \beta(A - q_1 - q_2^*))q_1 \\ &= \beta\left(A - q_1 - q_2^*\right))q_1 \\ &= \beta\left(A - q_1 - \frac{A - q_1}{2}\right))q_1 \\ &= \frac{\beta}{2}\left(A - q_1\right)q_1 \end{aligned}$$

- Hence optimal quantities: $q_1^* = A/2$ and $q_2^* = A/4$.
- Notice: optimal total quantity $q_T^* = q_1^* + q_2^* = (3/4)A$

Stackelberg (sequential) versus Cournot (simultaneous)

So compared to Cournot, in Stackelberg competition:

- the leader produces more and has higher profits ($u_1^* = (1/8)A^2\beta = 2u_2^*$)
- the follower produces less and has lower profits ($u_2^* = (1/16)A^2\beta$)

This is called a **first mover's advantage**:

- The leader (first mover) gets to optimise its utility (*anticipating BR from firm 2*);
- The follower just does the best it can (best response);
- Notice that the leader is **not** playing BR to the follower!

Backward induction and Nash eq.

- Backward induction results in a Nash eq.
- There are additional Nash eq., however they are based on non-credible threats, i.e. choices in the game-tree (i.e. in subgames) that are not rational.
- Nash eq. that depend on non-credible threats should be eliminated.
- This gives rise to the concept of **subgame perfect (Nash) equilibrium**
- Convert sequential game to normal form: Often easier to check.

From Extensive for Normal Form (perfect information)

Aim: Transform sequential game in normal form to use standard methods to find NEs.

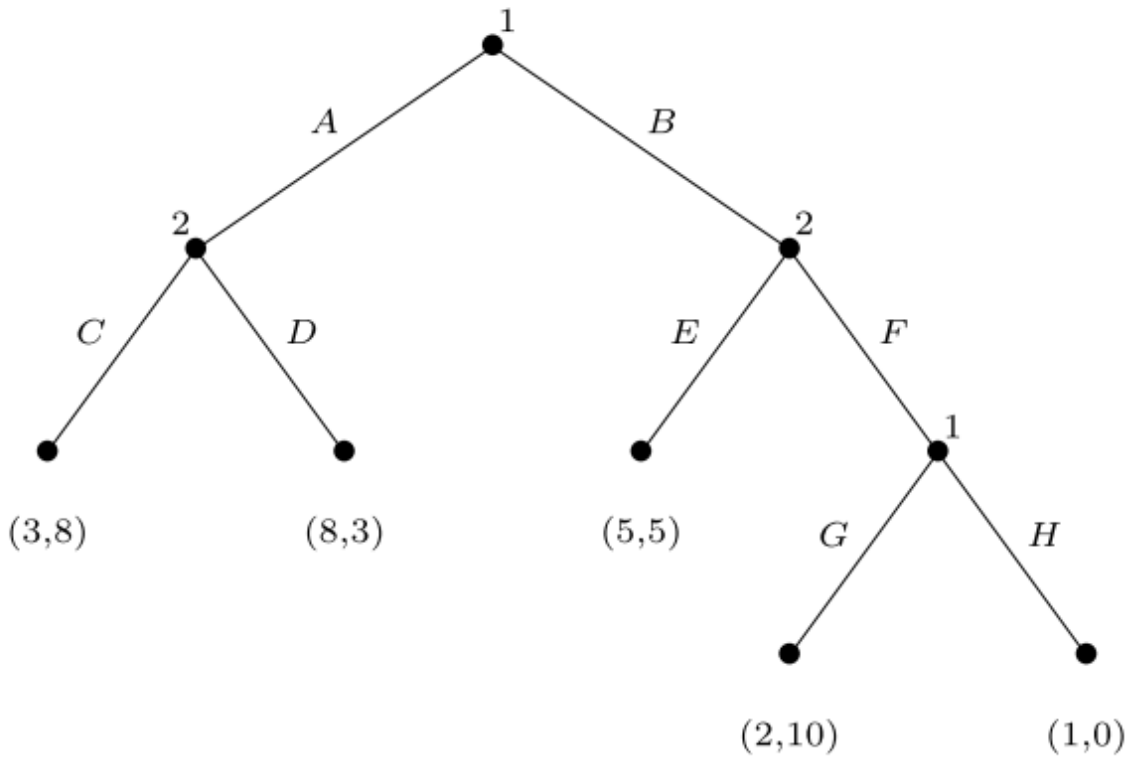


Figure 5.2: A perfect-information game in extensive form.

From Extensive to Normal Form (perfect information)

- **A pure strategy** for player i in a (perfect information) sequential game is a complete plan of action specifying which action to take at each of its decision nodes ...
- ... irrespective of **whether or not that node can be reached** when playing the strategy!
- Mathematically: it's the **product space** of the possible actions in each decision node:
  - Node 1 has 2 decision nodes: hence $\{A, B\} \times \{G, H\} = \{(A, G), (A, H), (B, G), (B, H)\}$
  - Node 2 has 2 decision nodes: hence $\{C, D\} \times \{E, F\} = \{(C, E), (C, F), (D, E), (D, F)\}$

From extensive to normal form

- **Alternative perspective:**
  1. Extensive form: player "waits" till one of his nodes is reached, then decides what to do;
  2. Normal form: each player makes a complete contingent plan in advance.
- **Informally:**
  - It's a **complete and contingent plan** instructing an assistant playing on your behalf, what to do in **each possible situation;**
  - Suppose that your assistant misunderstood and ended up in another node, then he still needs to know what to do.
  - Allows to explore whether unilateral deviation would be advantageous (Nash criterion)

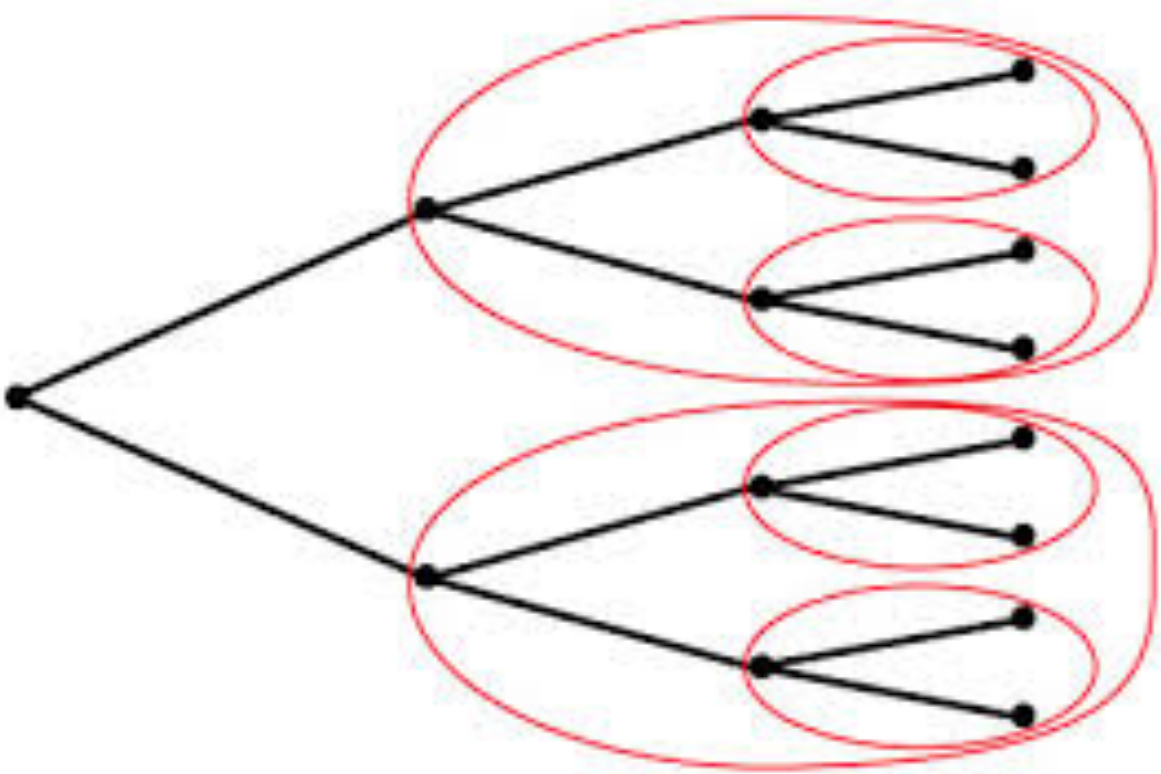
Determining Nash eq. in Normal Form

	(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 5.4: Equilibria of the game from Figure 5.2.

Introducing Subgames

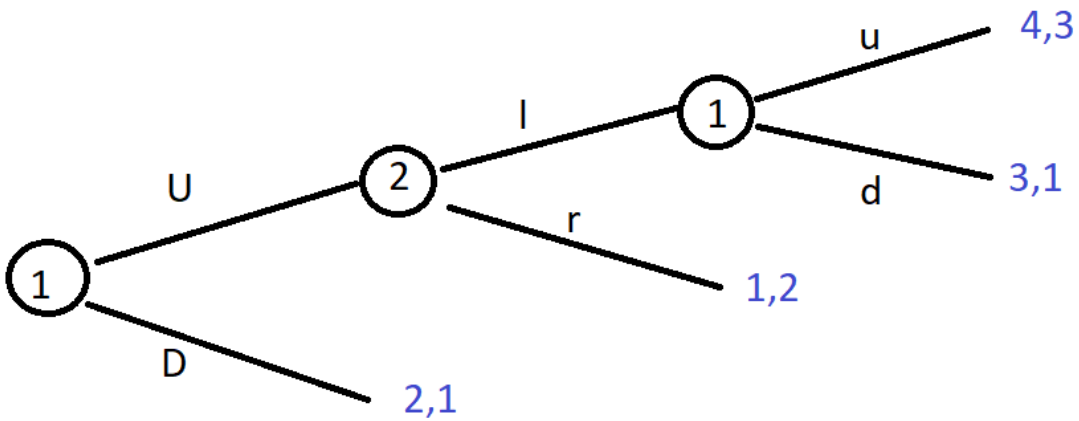
- Intuitively, subgames are subtrees of the game tree.



Introducing subgames for subgame perfection

- **Why subgames?** Allows us to translate the notion of non-credible threats to the normal form.
- **Continuation strategies** are a restrictions of a strategy for the original game to all of its subgames.
- **Subgame perfect Nash equilibrium (SPNE)** A strategy profile for an extensive-form game is a SPNE if it specifies a Nash equilibrium in each of its subgames.
- **SPNE** requires that what the players would do, conditional on being dropped in any of the subgames. should constitute a NE, even if a strategy profile dictates that certain subgames will not reached!
- This off-the-equilibrium-path behavior can be important, because it affects the incentives of players to follow the equilibrium.

Example: Backward Induction vs. Nash Equilibria



	ℓ	r	
Uu	$\underline{4,3}$	$1,2$	backward induction!
Ud	$3,1$	$1,2$	
Du	$2,1$	$\underline{2,1}$	"r non-credible threat"
Dd	$2,1$	$\underline{2,1}$	"d non-credible threat"

Non-credible threats do not constitute NE in their subgame!

Subgame Perfect Nash Equilibrium

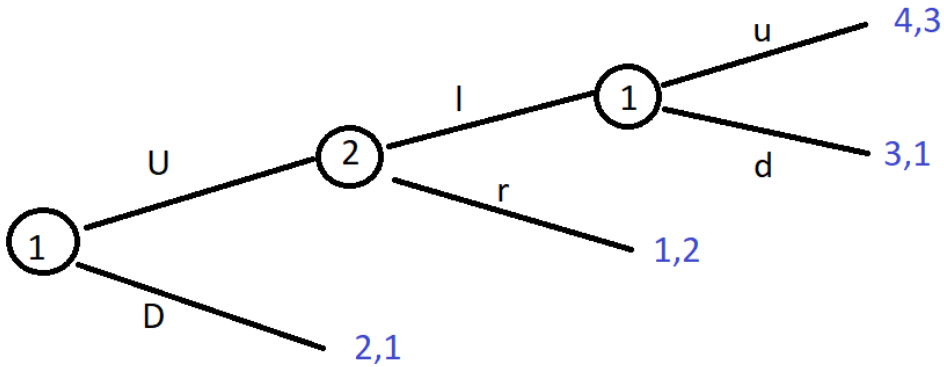
SGPE/SPNE : refinement of Nash Equilibrium:

Subgame-perfect equilibrium (SPGE, Selten 1965)

A Nash equilibrium s (of game G as a whole) is **subgame-perfect** iff for every subgame G' of G, the restriction of s to G' is also a Nash equilibrium.

- SGPE rules out Nash equilibria that rely on non-credible threats;
- Put differently: SGPE is the study of credible threats.

Example 2: Nash equilibria for subgames



	ℓ	r	
Uu	$\underline{4,3}$	$1,2$	backward induction!
Ud	$3,1$	$1,2$	
Du	$2,1$	$\underline{2,1}$	"r non-credible threat"
Dd	$2,1$	$\underline{2,1}$	"d non-credible threat"

$SG2:$

	ℓ	r
u	$\underline{4},\underline{3}$	$\underline{1},2$
d	$3,1$	$\underline{1},\underline{2}$

Notes

- d non-credible threat in SG1, implies r non-credible threat in SG2;
- Subgames (e.g. SG2) can have extra NE (comp. to full game)

Example 2, continued

Game has two non-trivial subgames:

- SG1 rooted at 2nd decision node of 1,
- SG2 rooted at decision node of 2

Normal form yields 3 NE's. Do they induce NE in all subgames?

- $NE1 = (Dd, r)$ with utility $(2, 1)$:
  induces action d in SG1 (not NE!)
- $NE2 = (Du, r)$ with utility $(2, 1)$:
  induces action (u, r) in SG2 (not NE!)
- $NE3 = (Uu, \ell)$ with utility $(4, 3)$:
  induces actions u in SG1, (u, ℓ) in SG2 (OK!)

Finding SGPE in perfect information games

Two approaches:

- **Backward induction (See next slide)**
  - works if NO infinite horizons!
- **Matrix form**
  1. Convert game-tree into matrix;
  2. Find all Nash equilibria;
  3. Eliminate the ones that depend on non-credible threats, i.e. do not induce a NE for each subgame;

Backwards induction

Algorithm to find subgame perfect equilibrium

- Consider each subgame of the game (in increasing order of inclusion)
- Find the NE for the subgame;
- Replace the subgame by a new terminal node that has the equilibrium pay-offs;

Zermelo's Thm (1913)

- With perfect information (one player in each iteration), a deterministic move is optimal. Hence there is a SGPE where each player uses a pure strategy.
- **For games with imperfect information**, a SGPE may require mixed strategies.

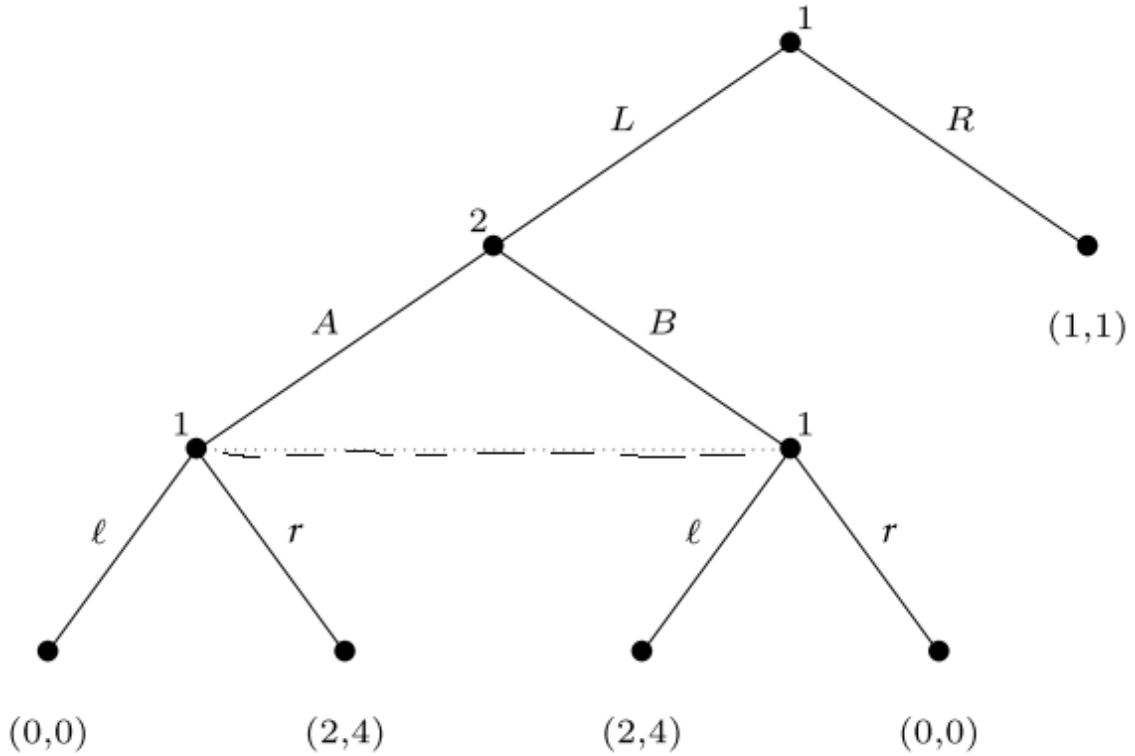
Imperfect information and information sets

- **Imperfect information: Intuition** Players need to act
  - with partial or no knowledge of actions taken by others,
  - with partial recall, i.e. limited memory of own past actions.

- An **imperfect- information game** is an extensive-form game in which each player's decision nodes are partitioned into **information sets;**

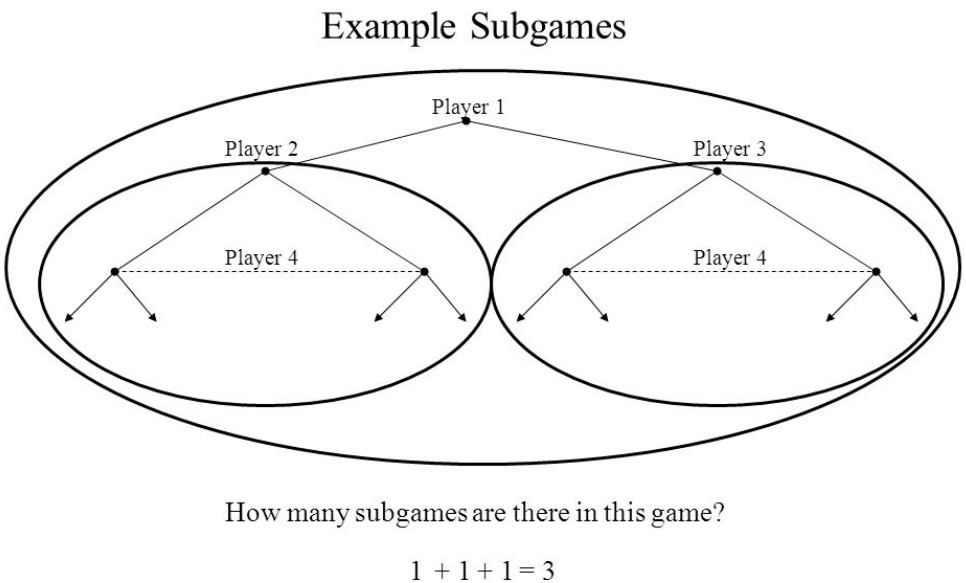
- Intuitively, if two decision nodes are in the same information set then the agent cannot distinguish between them.

Subgames: Condition on information set



- at any information set, a player must have the same strategies regardless of how the player arrived there;
- Subgames cannot break up information sets!

Subgame of a Sequential Game with Imperfect Information



Subgame definition:

- SG's initial node has singleton informationset;
- All successors are in SG;
- Any information-set is either completely in or out;

Pure strategies and induced normal form

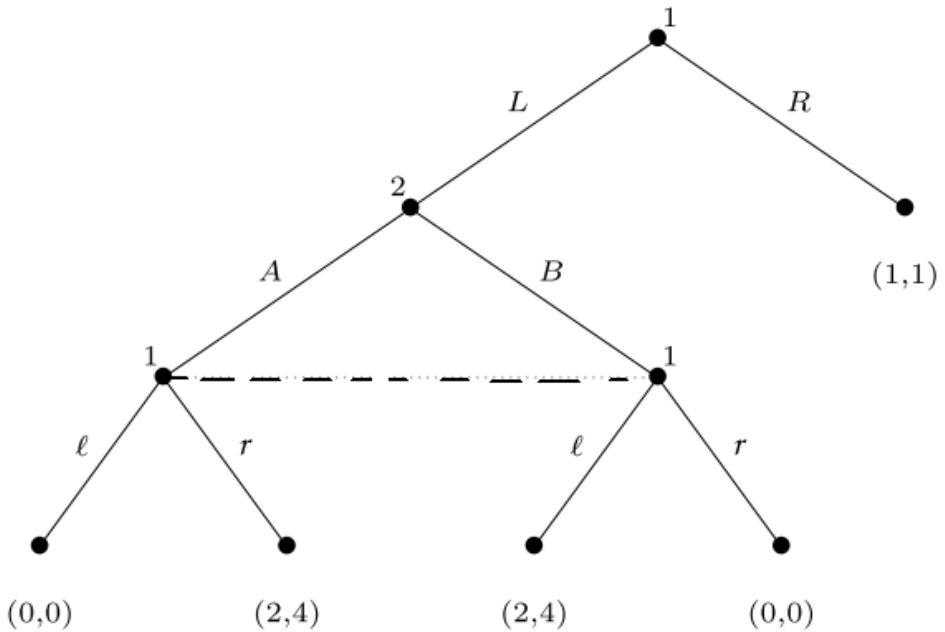


Figure 5.10: An imperfect-information game.

	A	B
$L\ell$	$0,0$	$2,4$
Lr	$2,4$	$0,0$
$R\ell$	$1,1$	$1,1$
Rr	$1,1$	$1,1$

Figure 5.14: The induced normal form of the game from Figure 5.10.

Pure actions are cartesian products over actions in information sets.

Subgame Perfect Nash Equilibrium for Imperfect Information Games

SGPE (aka SPE/SPNE) : refinement of Nash Equilibrium:

Subgame-perfect equilibrium (SPGE, Selten 1965)

A Nash equilibrium s (of game G as a whole) is **subgame-perfect** iff for every subgame G' of G, the restriction of s to G' is also a Nash equilibrium.

- SGPE rules out Nash equilibria that rely on non-credible threats;
- Put differently: SGPE is the study of credible threats.

Generalized Backwards Induction for Imperfect Information Games

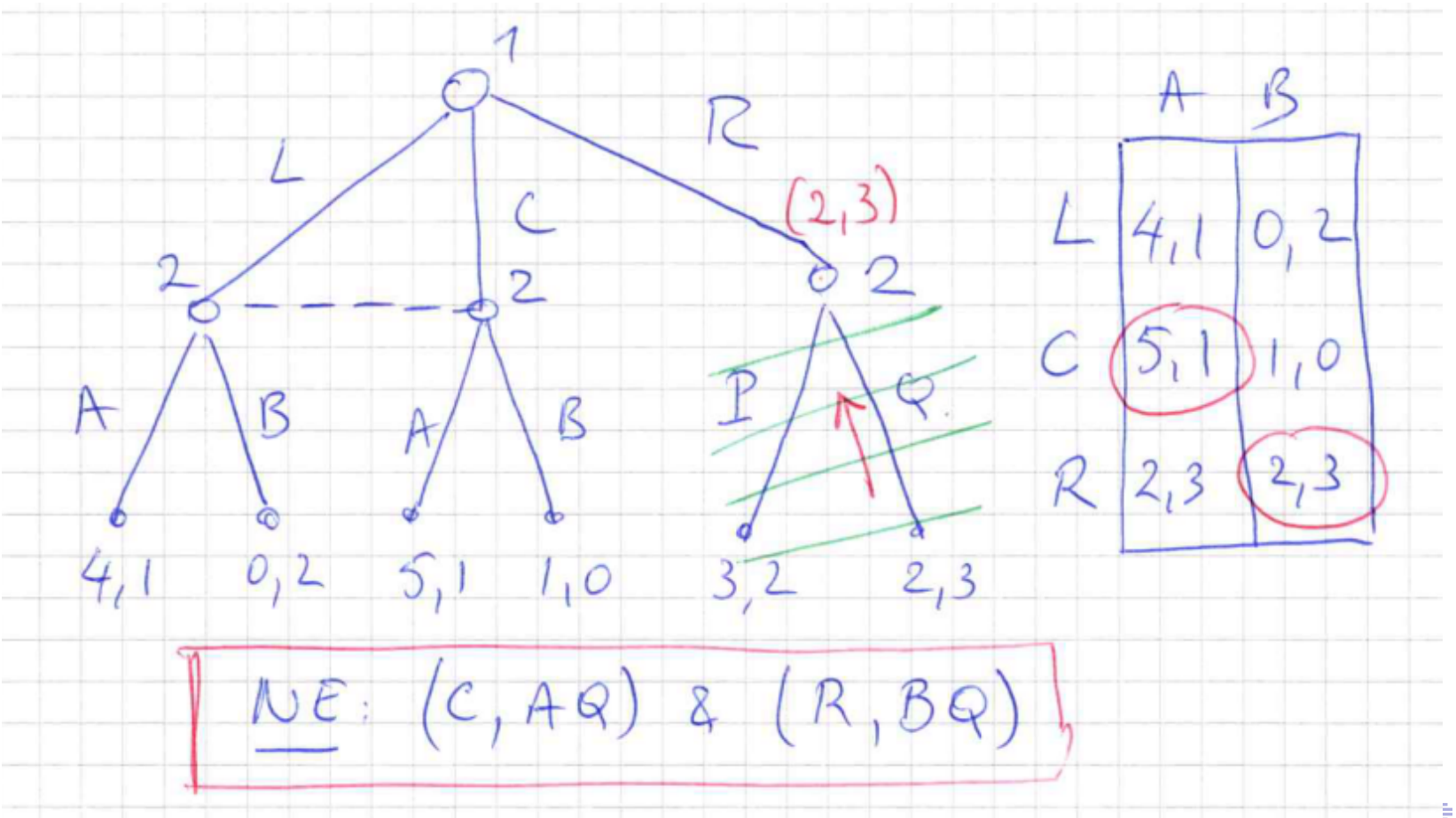
Systematically proceed as follows (if possible)

- Consider in turn each subgame of the game in increasing order of inclusion)

  Start at end of game, (no strategic interaction left)
  work backwards to beginning!
- Apply BI as far as you can;
- Replace the subgame by a new terminal node(s) that has the equilibrium pay-offs
  (we might need to consider different possibilities);
- If BI is not possible, use normal form solution techniques to find NE(s) for the remaining game (including mixed ones);

Generalized BI: Example 1, continued

- Not all "silly" NE (e.g. $(2, 3)$) can be eliminated (not enough subgames)
- Need for extra refinement (outside scope of this course).



Backward Induction vs. Subgame Perfection

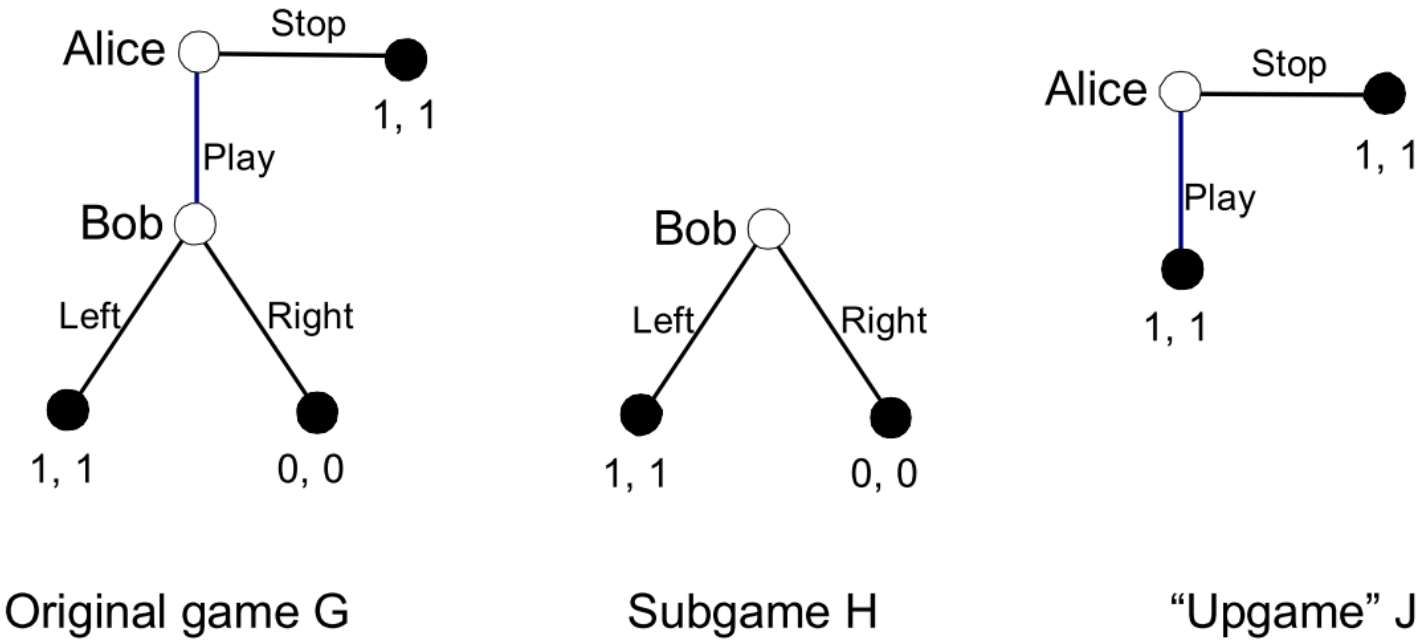


Figure 1. Backward induction versus subgame perfection.

The backward induction "upgame" J is NOT a subgame!

Ref: M. Kaminski: Generalized Backward induction, *Games* 2019, 10, 34

Backward Induction vs. Subgame Perfection

Finite sequential games with perfect information:

- All SPE can be found by backward pruning, ie.
  - systematic and incremental substitution of terminal subgames with Nash-eq pay-offs
- All BI solutions (backward pruning) are SPE

Backward Induction also works in some more general cases

e.g. some infinite games (e.g. Rubinstein)

More complex games require more restrictions on the Nash eq. solution to eliminate unreasonable solutions.

E.g. sequential rationality, perfect equilibrium

The Ultimatum Game

Ultimatum Game (UG): baseline (simplest non-trivial) model for **bargaining:** *take-it-or-leave-it!*

Assumptions

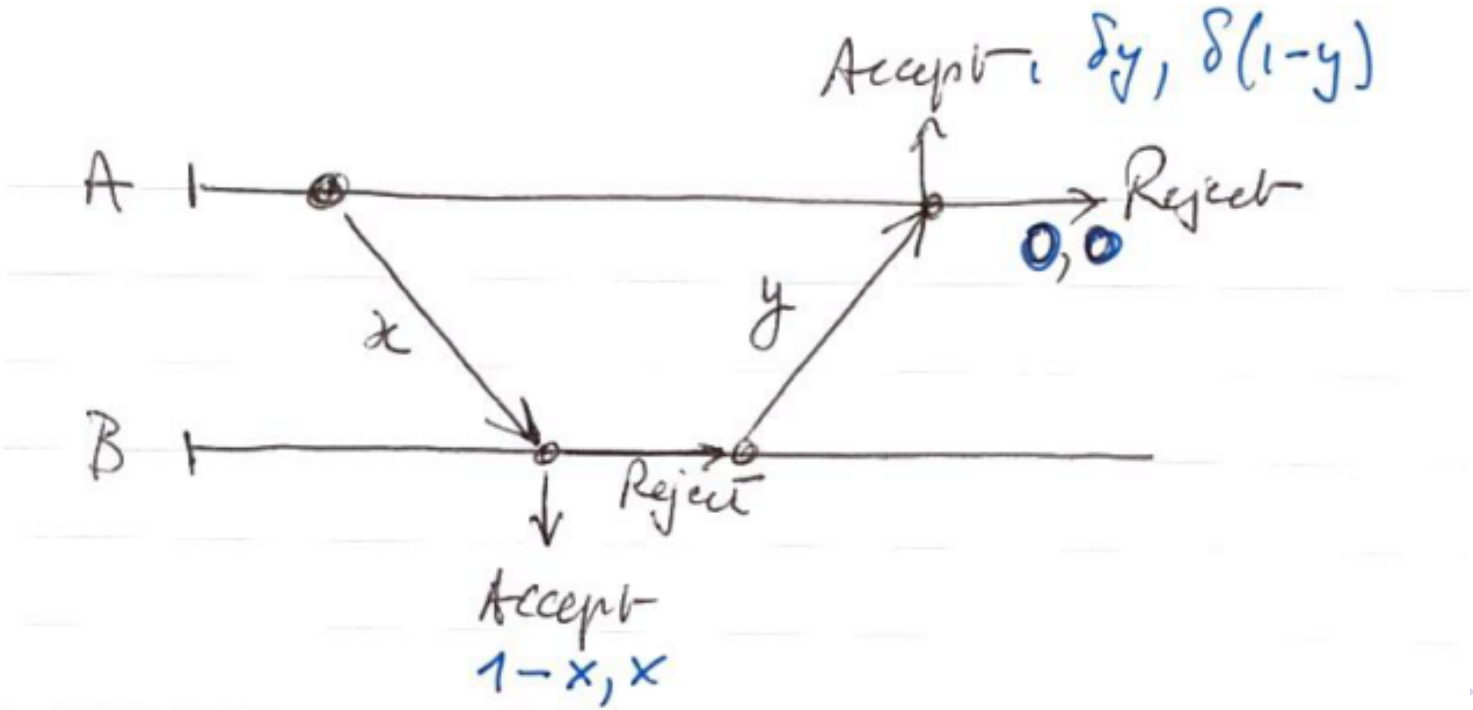
- **Surplus** can be divided continuously ($0 \leq x \leq 1$)
- **Two agents:**
  - A **proposes** split x versus $1 - x$, (**proposal power**)
  - B accepts or rejects;
- **No deal (conflict deal)** is considered **worst outcome**;
- Both agents aim to **maximize their utility**;

Two rounds ultimatum game with impatient players

Power of counter-offer:

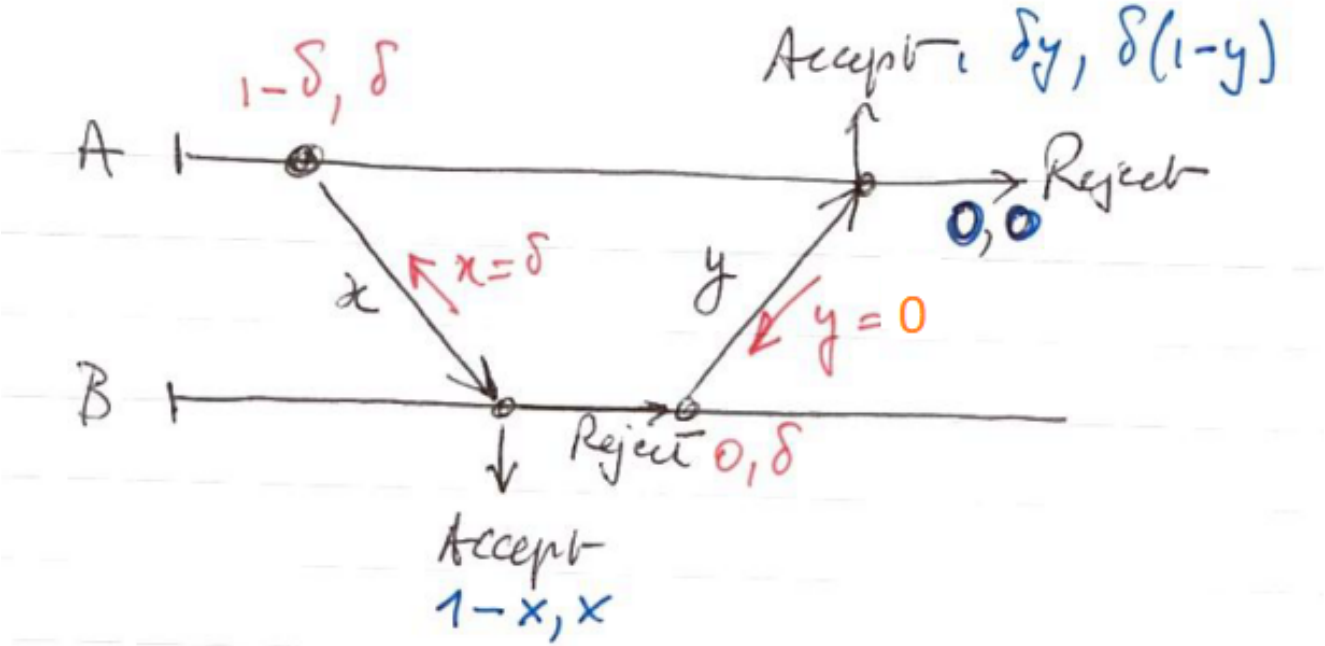
- A makes offer;
- B either accepts, or makes counter-offer.
- However, in each round the total is reduced by factor $\delta < 1$ (the icecream is melting!)

A makes offer, but B can make counter-offer!



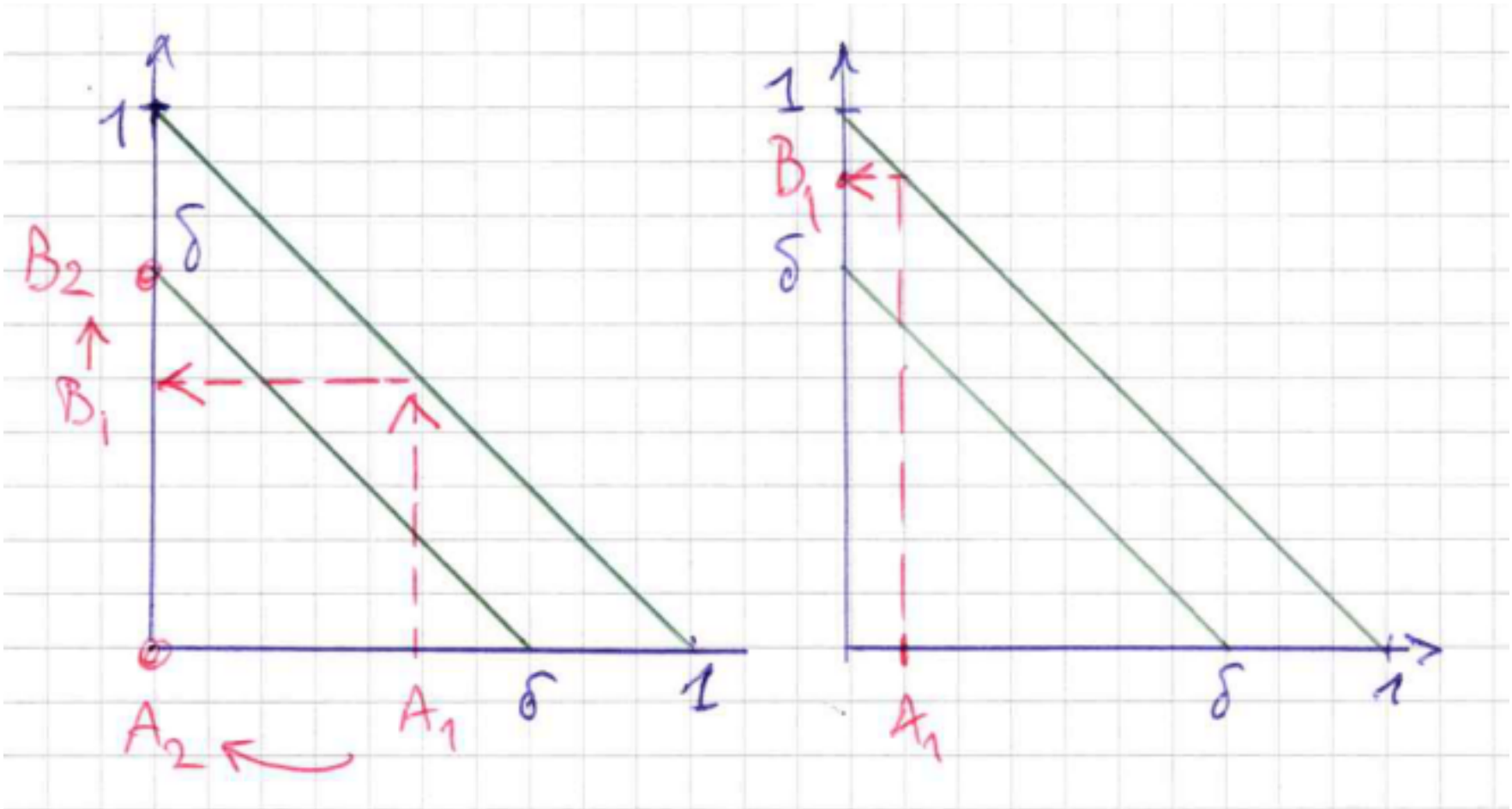
Two rounds ultimatum game: BI solution

- A makes offer, but B can make counter-offer!
- Use **backward induction** to find optimal solution.

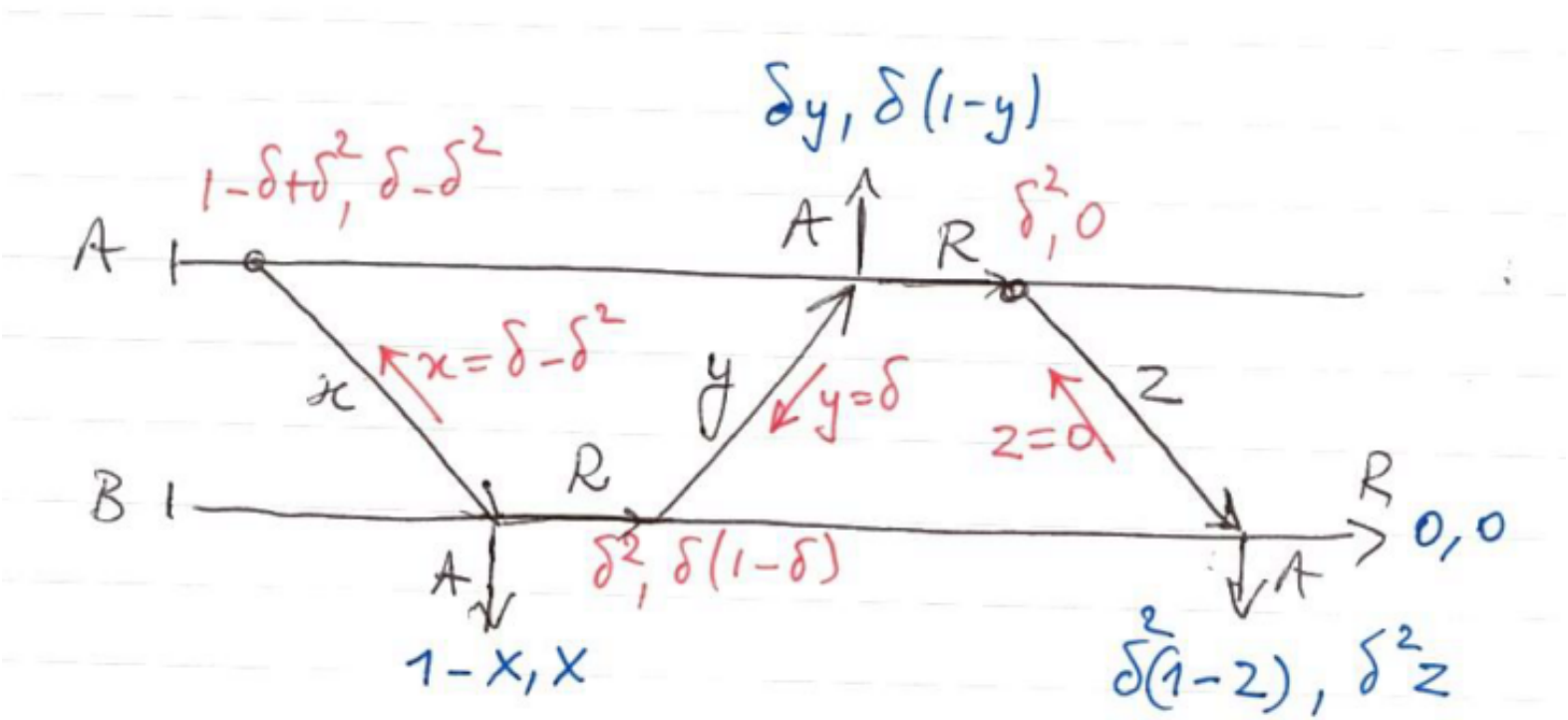


Conclusion: A offers split $(1-\delta, \delta)$ which B accepts.

Alternative interpretation



Three rounds ultimatum game: Backwards induction



Conclusion: A proposes split $(1-\delta+\delta^2, \delta-\delta^2)$ which B accepts.

Alternating offers bargaining

- Generalizing to negotiation with n rounds;
- When number of negotiation rounds is fixed at n, the equilibrium split is given in table below:

payoff for n rounds	$n=1$	$n=2$	$n=3$	$n=4$
for A	1	$1-\delta$	$1-\delta+\delta^2$	$1-\delta+\delta^2-\delta^3$
for B	0	δ	$\delta-\delta^2$	$\delta-\delta^2+\delta^3$

Mathematical aside

$$\sum_{k=0}^{\infty} x^k = 1 + x + x^2 + x^3 + \ldots = \frac{1}{1-x} \qquad (\text{for } |x| < 1)$$

$$\sum_{k=1}^{\infty} x^k = x + x^2 + x^3 + x^4 \ldots = \frac{x}{1-x} \qquad (\text{for } |x| < 1)$$

$$\begin{aligned} 1 + x + x^2 + \ldots + x^n &= \frac{1}{1-x} - x^{n+1}\frac{1}{1-x} \\ &= \frac{1 - x^{n+1}}{1-x} \end{aligned}$$

General conclusions

- Limit behaviour for pay-offs:

$$A(n) = 1 - \delta + \delta^2 - \ldots \pm \delta^{n-1} = \frac{1-(-\delta)^n}{1-(-\delta)}$$

$$\Longrightarrow \lim_{n\to\infty} A(n) = \frac{1}{1+\delta}$$

$$B(n) = 1 - A(n) \quad \Longrightarrow \quad \lim_{n\to\infty} B(n) = \frac{\delta}{1+\delta}$$

- **First offer advantage:** $\lim A(n) \geq \lim B(n)$
- First offer advantage disappears for very patient negotiators:

$$\delta \longrightarrow 1 \quad \Longrightarrow \quad A(n) \searrow 1/2, \quad B(n) \nearrow 1/2;$$

Rubinstein's Model: Infinite Horizon Bargaining

- 2 agents, infinite horizon (# rounds **not fixed in advance**);
- **Time is valuable:** discount factors for both agents (δ_1, δ_2);
- **Optimal split:**

$$u_1 = \frac{1-\delta_2}{1-\delta_1\delta_2} \qquad \text{and} \qquad u_2 = \frac{\delta_2(1-\delta_1)}{1-\delta_1\delta_2}$$

- **Tragedy of bargaining:**
  The more time matters, the lower the share!
  E.g. if $\delta_2 \approx 0$, then $u_2 \approx 0$, etc.
- Notice **first mover's advantage** for $\delta_1 = \delta_2 = \delta$:

$$u_1 = \frac{1-\delta}{1-\delta^2} = \frac{1}{1+\delta} \qquad \text{and} \qquad u_2 = \frac{\delta(1-\delta)}{1-\delta^2} = \frac{\delta}{1+\delta}$$

Repeated Games

- Repeat the **same** one-shot (stage) game (special case of sequential game);
- At each stage, information about preceding games is known;
- **Finite number** n (known!) of repetitions;
  - Maximize total reward:

$$R_n = \sum_{t=1}^{n} r_t$$

- **Infinite (unlimited) number** of repetitions
  - Maximize **discounted** total reward:

$$R = \sum_{t=1}^{\infty} \delta^{t-1} r_t \qquad \text{discount factor} \quad 0 < \delta < 1$$

  - equivalently: ending after **random number** of repetitions

Repeated games: interpretation of discount factor

- After every stage-game there is a
  - probability $1-\delta$ that game will be ended;
  - probability δ that game will proceed to next round

- Reward R now becomes random variable:

$$R_N = \sum_{t=1}^{N} r_t \qquad \text{where} \quad P(N=n) = \delta^{n-1}(1-\delta)$$

(Assuming at least one stage game is played, hence $N \geq 1$).

Then:

$$E(R_N) := E\left(\sum_{t=1}^{N} r_t\right) = r_1 + \delta r_2 + \delta^2 r_3 + \ldots = \sum_{t=1}^{\infty} \delta^{t-1} r_t.$$

Example: Repeated Prisoner's dilemma

- Pay-off matrix for **stage game:**

	$C(oop)$	$D(efect)$
C	$3, 3$	$1, 4$
D	$4, 1$	$2, 2$

- For **finite number** of repetitions:
  - Single Nash eq.: Play D-D for each repetition;
  - Can be proved using **backward induction**
- For **infinite number** of repetitions:
  - More interesting strategies, including cooperation
  - Examples: Tit-for-Tat, Grim Trigger

Grim Trigger Strategy for Repeated Prisoner's Dilemma

- **Grim Trigger and Tit-for-Tat strategy**
  Start by cooperating ...
  - GT: continue cooperating, until someone defects; from then onwards, always defect!
  - TfT: from then onwards, copy last move of opponent.
- If both parties play GT, is it rational to defect?
  Consider player 1:
  - Utility for continued **cooperation:**

$$u_1(C) = 3 + 3\delta + 3\delta^2 + \ldots = \frac{3}{1-\delta}$$

  - Utility for **defection**:

$$u_1(D) = 4 + 2\delta + 2\delta^2 + \ldots = 4 + 2\frac{\delta}{1-\delta}$$

- Continued **cooperation is rational** if $u_1(C) > u_1(D)$.

Grim Trigger Strategy for Repeated Prisoner's Dilemma

- Continued **cooperation is rational** if $u_1(C) > u_1(D)$:

$$u_1(C) > u_1(D) \iff \frac{3}{1-\delta} = 4 + 2\frac{\delta}{1-\delta}$$

$$\iff \delta > 1/2.$$

- **Interpretation:** If the players are sufficiently patient (i.e. future rewards are sufficiently valuable) then it's rational to cooperate.

Coalitional Game Theory

- Basic modelling unit is **group** rather than indvidual agent.
- Transferable vs. non-transferable utility
- **Coalitional game with transferable utility** (N, v)**:**
  - N finite set of players:
  - $v : 2^N \longrightarrow \mathbb{R}$ pay-off function $(v(\emptyset) = 0)$
- Fundamental questions:
  - Which coalitions will form?
  - How should coalition divide its pay-off among its members?

Classes of coalitional games

- **Super-additive game ("synergy")**
  Game (N, v) is super-additive iff

$$\forall\, S, T \subset N : S \cap T = \emptyset \Longrightarrow v(S \cup T) \geq v(S) + v(T).$$

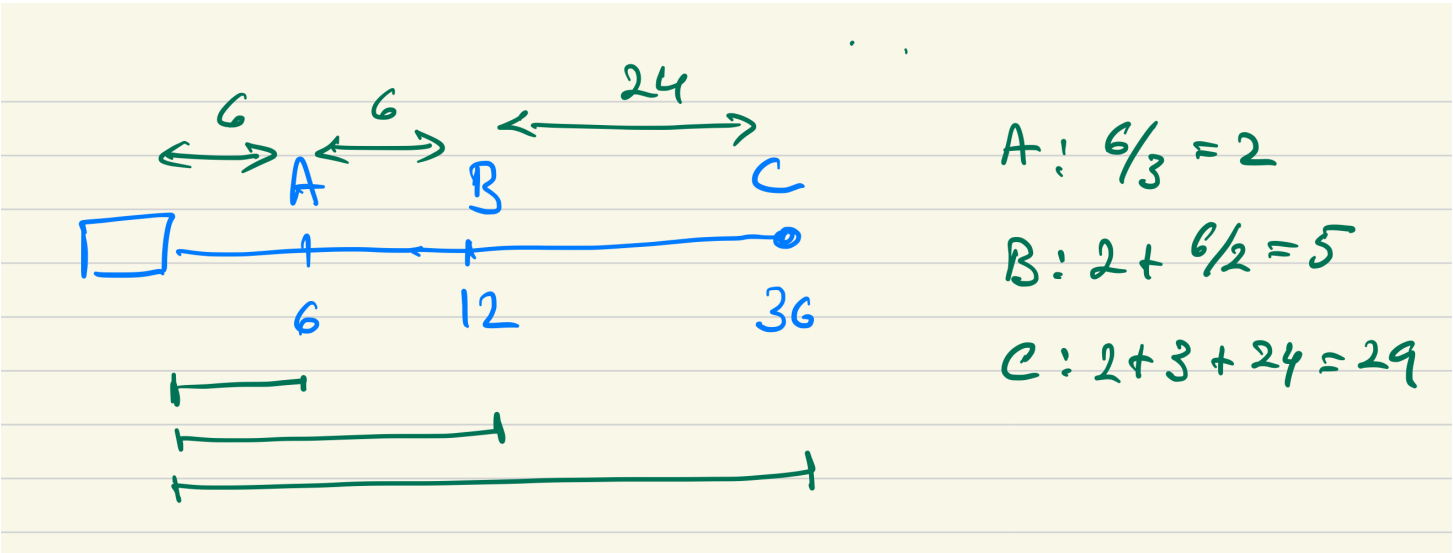
  In particular: $v(S \cup i) \geq v(S) + v(i)$ for any $S \subset N \setminus \{i\}$.
- As a consequence, for super-additive game, the grand coalition has the highest pay-off of all coalitional structures:

$$v(N) = v(S \cup S^c) \geq v(S) + v(S^c) \geq v(S).$$

- Therefore focus on a fair redistribution of total pay-off among the members of the grand coalition.

Worked example: Fair division of taxi fare

- Alice, Bob and Charlize share a taxi to go home;
- The individual fares would be: A(6), B(12) and C(36);
- If they share the cab then they only have to pay the fare to the farthest destination (C = 36).
- What would be a fair way to share the fare?



Sanity check: 2+ 5+29 = 36, and everyone is better off!

Worked example: Fair division of taxi fare

- Consider a sequential version of the problem in which A, B and C arrive in random order, and pay whatever is lacking (i.e. their marginal contribution);
- Permutation ACB indicates that coalition grows as follows:

$$A \rightarrow AC \rightarrow ACB$$

1. When A joins, he pays the fare to his destination: 6
2. When C joins, he pays the remainder to get to C: 36-6=24;
3. Finally, when B joins, everything is already paid for.

Shapley value: Alternative definition

- Marginal contribution only depends on what precedes a contributor;
- Marginal contribution of player i to subset S:

$$\delta_i(S) = v(S \cup i) - v(S)$$

- Shapley value of player i: (denoting $\#N = n, \#S = s$)

$$\varphi_i(N, v) := \frac{1}{n} \sum_{S \subset N \setminus i} \binom{n-1}{s}^{-1} \delta_i(S)$$

- Amplification: see next slides!

Shapley Value: Amplification

- We focus on Shapley value $\varphi_i(N, v)$ for agent i;
- For any existing coalition S not including i, i.e.

$$S \subset N_i := N \setminus i$$

we consider the value increment due to i joining:

$$\delta_i(S) = v(S \cup i) - v(S)$$

- The size $s := \#S$ of the possible coalitions S that i joins, can range between $0 \leq s \leq n - 1$.

Shapley Value: Amplification

- For fixed coalition size s there are

$$N_s := \binom{n-1}{s}$$

coalitions S of that size.

- Hence, the mean contribution $\overline{\Delta}_i$ of i to existing coalitions S of size s is given by:

$$\overline{\Delta}_i(s) := \frac{1}{N_s} \sum_{S:\#S=s} \delta_i(S) = \binom{n-1}{s}^{-1} \sum_{S:\#S=s} \delta_i(S).$$

Shapley Value: Amplification

- Finally, since $0 \leq s \leq n-1$ we compute the average over the n possible choices of s. This average is the Shapley value:

$$\begin{aligned}
\varphi_i &:= \frac{1}{n}\sum_{s=0}^{n-1} \overline{\Delta}_i(s) = \frac{1}{n}\sum_{s=0}^{n-1} \binom{n-1}{s}^{-1} \sum_{S:\#S=s} \delta_i(S) \\
&= \frac{1}{n}\sum_{s=0}^{n-1} \sum_{S:\#S=s} \binom{n-1}{s}^{-1} \delta_i(S) \\
&= \frac{1}{n}\sum_{S\subset N\setminus i} \binom{n-1}{s}^{-1} \delta_i(S)
\end{aligned}$$

- Double sum above is actually sum over all subsets $S \subset N \setminus i$.

Shapley Value: Amplification

Recall:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \quad \text{where} \quad n! = n(n-1)(n-2)\ldots 3 \cdot 2 \cdot 1.$$

Hence:

$$\begin{aligned}
\varphi_i &= \frac{1}{n} \sum_{S \subset N \setminus i} \binom{n-1}{s}^{-1} \delta_i(S) \\
&= \frac{1}{n} \sum_{S \subset N \setminus i} \frac{s!(n-1-s)!}{(n-1)!} \delta_i(S) \\
&= \frac{1}{n!} \sum_{S \subset N \setminus i} s!(n-1-s)!\, \delta_i(S)
\end{aligned}$$

Claim: this corresponds to the permutation definition!

Shapley: equivalence of definitions

Recall: $\delta_i(S)$ depends on the **set** S, not on the permutation sequence of the elements in S!

- Hence, $\delta_3(S)$ has the same value in $\pi_1 = 24315$ and $\pi_2 = 42351$ as in both cases $S = \{2, 4\}$.
- Consider an arbitrary permutation of $1, 2, \ldots, n$ and let's focus on i somewhere in the sequence, all numbers that appear to the left of i, constitute the set S. Likewise, all elements that appear to the right, are collected in the set S^c:

$$\pi = \underbrace{* * * \ldots * **}_{S} \, i \, \underbrace{* * * \ldots * **}_{S^c}$$

- Any permutation of the S-elements in π yields the same value $\delta_i(S)$ (see above), There are $s!$ such permutations.

Shapley: equivalence of definitions

- Likewise, $\delta_i(S)$ does **not depend** on the elements in S^c. Any permutation of these elements in S^c yields that same value $\delta_i(S)$. There are $(n-1-s)!$ such permutations.
- Hence, we can conclude that of the $n!$ permutations of $\{1, 2, \ldots, n\}$, a total of $s!(n-1-s)!$ give rise to the same value $\delta_i(S)$, which only depends on the **set** S.
- Averaging over all possible choices for $S \subset N_i \equiv N \setminus i$ yields:

$$\frac{1}{n!} \sum_{S \subset N_i} s!(n-1-s)!\, \delta_i(S) = \phi_i(i)$$

Shapley's Axioms: Some useful terminology

- Players i and j are **interchangeable** if their contributions to every coalition (subset) S is exactly the same:

$$\forall\, S \subset N \setminus \{i,j\}: \quad v(S \cup i) = v(S \cup j)$$

- A player i is a **dummy player** if the amount he contributes to any coalition is exactly the amount he's able to achieve alone:

$$\forall S \subset N \setminus \{i\} : v(S \cup i) = v(S) + v(i)$$

Shapley's Axioms

- **Symmetry:** If i and j are interchangeable then:

$$\psi_i(N, v) = \psi_j(N, v).$$

- **Dummy Player:** will only get what he can achieve on his own:

$$\psi_i(N, v) = v(i).$$

- **Additivity:** Consider two games $G_1 = (N, v)$, $G_2 = (N, w)$ and assume that we play G_1 with probability p and G_2 with prob $q = 1 - p$.
  Then

$$\psi_i(N, pv + qw) = p\psi_i(N, v) + q\psi_i(N, w)$$

Shapley's theorem

Shapley (1951)

Given a coalitional game (N, v), the Shapley values $\varphi_i, i = 1, \ldots, n$ specifies the unique distribution of the total value $v(N)$ that is both

- efficient, i.e. $\sum_i \varphi_i = v(N)$
- satisfies Shapley's axioms,
  i.e. *Symmetry*, *Dummy Player* and *Additivity*.

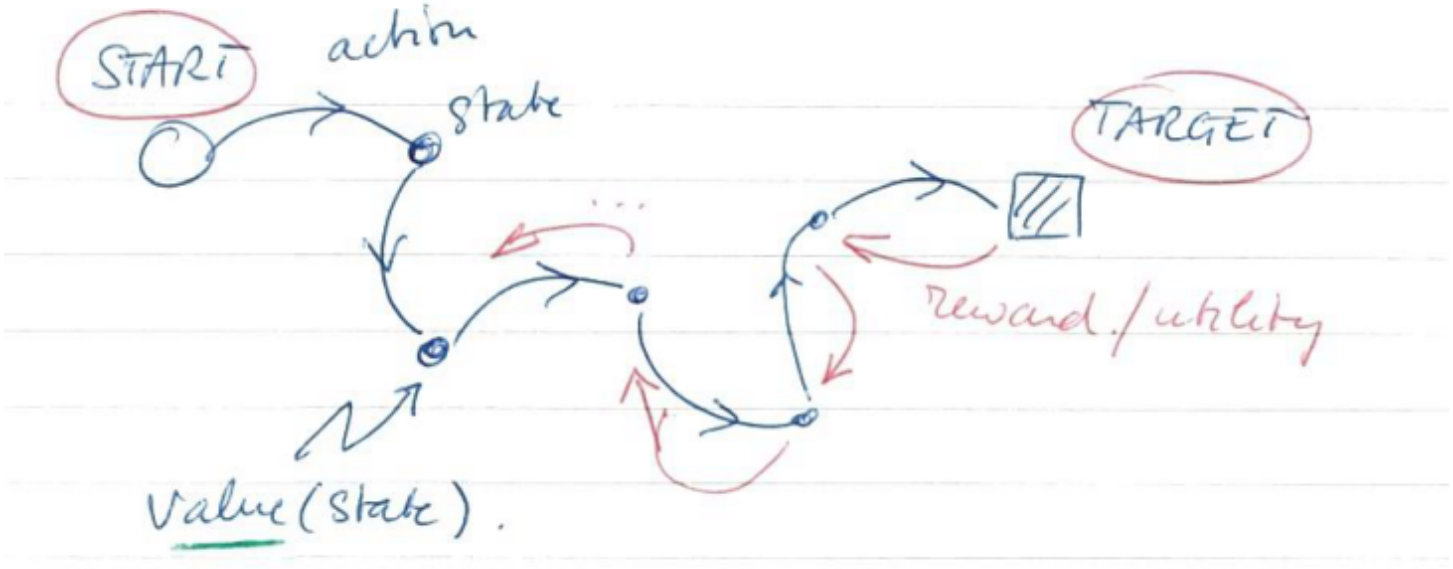
Shapley value: worked example

An AI expert (E) developed a powerful new algorithm. However, in order to implement his ideas, he needs to create a startup and hire a programmer (P) for 2 years. An angel investor (A) provides funding. The value that each coalition of these three stakeholders (E, P, A) can generate satisfies the following rules:

- Without both investor and expert, no value can be generated.
- If he has no assistance from a programmer, the expert's value equals 3, but if he can delegate the programming and focus on R&D, his value rises to 10.
- The value created by the programmer is 5. This is in addition to the rise in value of the expert.

The startup is sold to a large software company for serious money. How to split this money fairly among the three stakeholder?

Recurring themes in (sequential) decision making



- States, actions, transitions, policy, value functions;
- Back-up, optimisation (planning and searching)

Sequential Decision Making

- In **sequential decision making** an agent tries to solve a sequential control problem by directly interacting with an unknown environment
- **Learning by trial and error** Agent tries out actions to learn about their consequences
- **Not supervised:** No examples of correct or incorrect behavior; instead only **rewards** for actions tried
- **Active learning:** agent has partial control over what data it will obtain for learning
- **On-line learning:** it must maximize performance during learning, not afterwards!

Monte Carlo: Sample-based computation

- Let $X_1, X_2, \ldots, X_n$ be a sample (i.i.d) from given (discrete/continuous) distribution;

$$EX = \left\{ \begin{array}{c} \sum_{k=0}^{\infty} x_k P(X = x_k) = \sum_{k=0}^{\infty} x_k p_k \\ \int x f(x) dx \end{array} \right\} \approx \frac{1}{n} \sum_{i=1}^{n} X_i$$

$$E\varphi(X) = \left\{ \begin{array}{c} \sum_{k=0}^{\infty} \varphi(x_k) p_k \\ \int \varphi(x) f(x) dx \end{array} \right\} \approx \frac{1}{n} \sum_{i=1}^{n} \varphi(X_i)$$

Kullback-Leibler Divergence

- Let p, q be probability distributions (cont. or discrete)

Kullback-Leibler divergence

$$KL(p||q) := \int p(x) \log \frac{p(x)}{q(x)} \, dx \geq 0$$

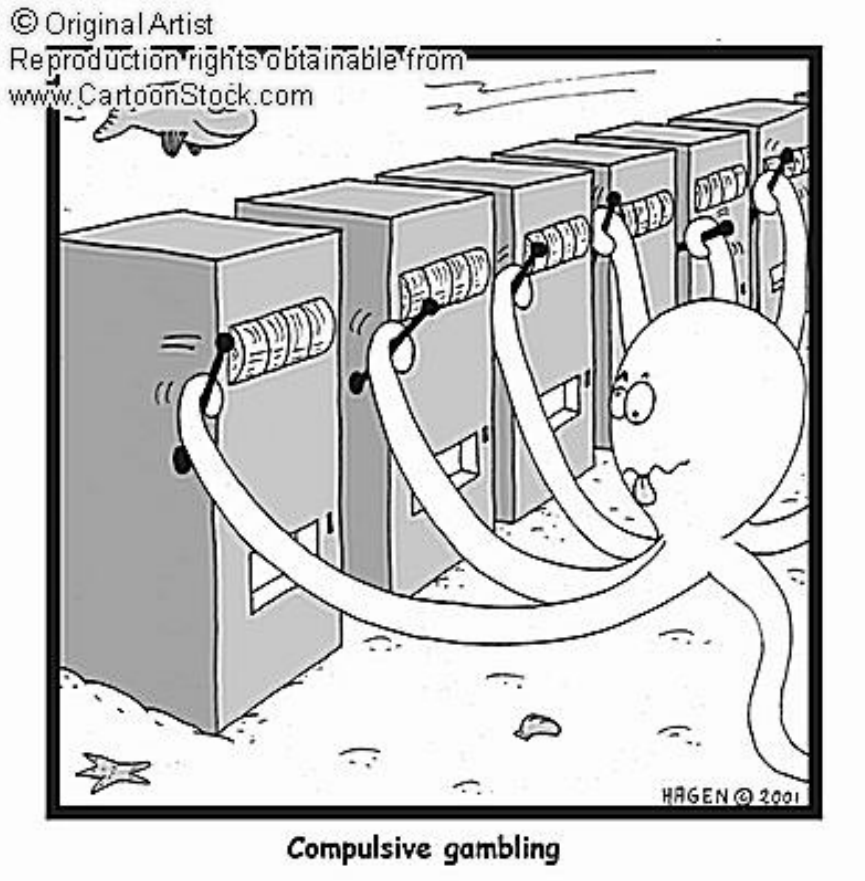
$$KL(p||q) := \sum_k p_k \log \frac{p_k}{q_k} \geq 0$$

- Divergence is **not symmetric:** $KL(p||q) \neq KL(q||p)$
- Sample $X_1, X_2, \ldots, X_n \sim p$:

$$KL(p||q) \approx \frac{1}{n} \sum_{i=1}^{n} \log \frac{p(X_i)}{q(X_i)}$$

Prototypical problem: K-armed bandit problem

- Sit before a slot machine (bandit) with many (k) arms;
- Each arm has an unknown stochastic payoff
- Goal is to
  - maximize cumulative payoff over some period
  - minimize regret?
  - others?



Possible goals in k-bandit problem

- Maximize optimal selection percentage
- Maximize cumulative/average payoff over some period
- Minimize regret
- Pure exploration: no cost associated with exploration, but decision is urgent!
- others?

Formalizing the k-armed bandit problem

- There are k **actions** available at each timestep;
- After action t, the agent receives (stochastic) reward $R_t \sim f_{a_t}$
- In a **finite-horizon** problem, the agent tries to maximize its **total reward** over T actions: $\sum_{t=1}^{T} R_t$
- In an **infinite-horizon** problem, the agent tries to maximize its **discounted total reward**: $\sum_{t=0}^{\infty} \gamma^t R_t$ where $0 < \gamma < 1$
- The **discount factor** γ can be interpreted as the probability of the game continuing after each step

Exploration and exploitation

The agent's ability to get reward in the future depends on what it knows about the arms.

- It must **explore** the arms in order to learn about them and improve its chances of getting future reward;
- It must use what it already knows in order to maximize its total reward; Thus it must **exploit** by pulling the arms it expects to give the largest rewards;

Balancing exploration and exploitation

- The main challenge in a k-armed bandit is how to **balance the competing needs of exploration and exploitation**
- If the horizon is finite, exploration should decrease as the horizon gets closer
- If the horizon is infinite but $\gamma < 1$, exploration should decrease as the agent's uncertainty about expected rewards goes down

Action-value methods

Formal Setup:

- Each arm (i.e. action $a = 1, \ldots, k$) generates stochastic reward R sampled from probability distribution f_a with unknown mean $q(a)$, hence: $q(a) := E_{f_a}(R)$
- **Action-value**: $q(a)$ can be thought of as the (average) value (quantity) generated by taking action a;
  - Compare to *state-action value* $q(s, a)$ in RL
- We use the sample average $Q_t(a)$ is an estimate of $q(a)$. Specifically, if action a has been chosen k_a times, yielding rewards $R_1, R_2, \ldots, R_{k_a}$, then:

$$Q_t(a) = \frac{1}{k_a} \sum_{i=1}^{k_a} R_i$$

Action-value methods

- Asymptotically:

$$Q_t(a) \longrightarrow q(a) \qquad \text{as} \qquad t, k_a \longrightarrow \infty.$$

- Incremental implementation if a is selected at time $t+1$:

$$\begin{aligned} Q_{t+1}(a) &= \frac{k_a Q_t(a) + R_{t+1}}{k_a + 1} \\ &= \frac{k_a}{k_a + 1} Q_t(a) + \frac{1}{k_a + 1} R_{t+1} \\ &= Q_t(a) + \frac{1}{k_a + 1}\Big[R_{t+1} - Q_t(a)\Big] \end{aligned}$$

- Example of an update rule for estimates:

$$NewEst \leftarrow OldEst + LearningRate[NewData - OldEst]$$

Defining exploration vs. exploitation

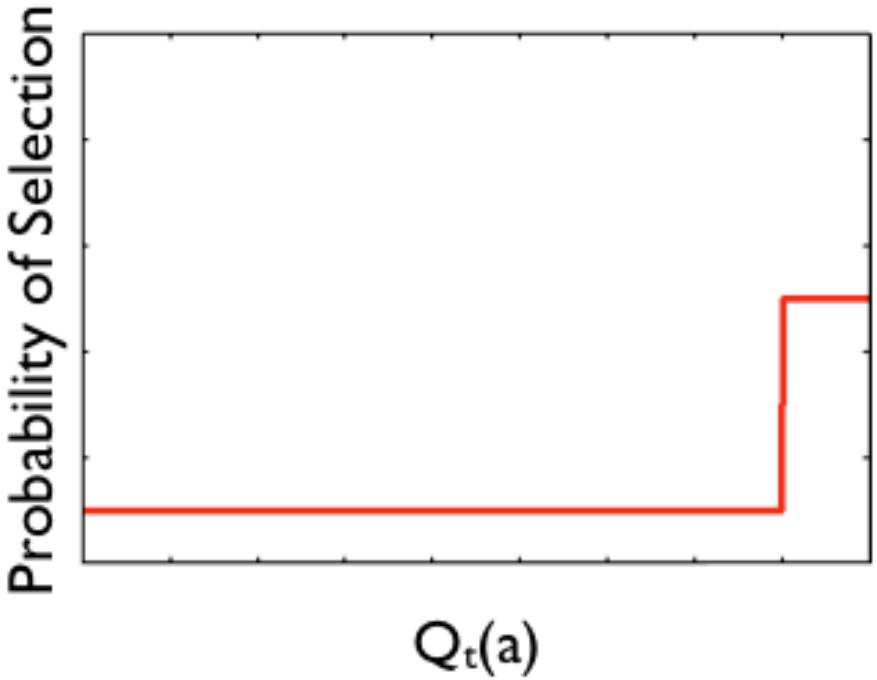
- When using action-value methods, exploration and exploitation are easy to define.
- **Exploiting** means taking the **greedy** action:

$$a^* = \arg\max_a Q_t(a)$$

- **Exploring** means taking any other action:
- Frequently used exploration strategies
  1. ϵ-greedy
  2. Soft-max

Epsilon-greedy exploration

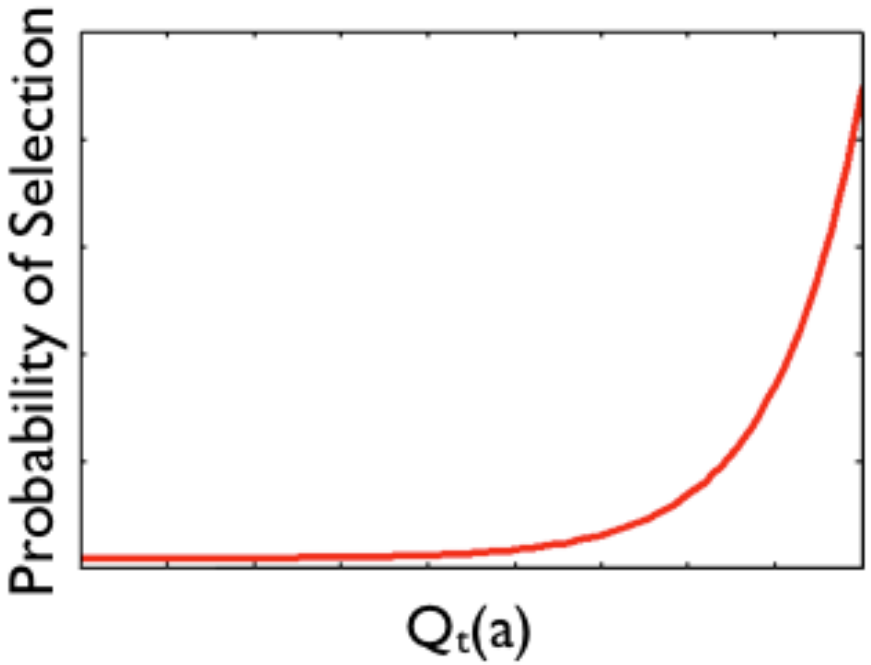
In ϵ-**greedy** exploration, the agent selects a random action with probability ϵ, and the greedy action otherwise



Softmax exploration

In **softmax** exploration, the agent chooses actions according to a **Boltzmann** distribution

$$p(a) = \frac{e^{Q(a)/\tau}}{\sum_{a'} e^{Q(a')/\tau}}$$



Temperature in softmax (Boltzman) distribution

Suppose $Q(a) > Q(b)$ then

$$\frac{p(a)}{p(b)} = \frac{e^{Q(a)/\tau}}{e^{Q(a)/\tau}} = e^{(Q(a)-Q(b)/\tau}$$

Hence

- If temperature increases ($\tau \uparrow$): distribution becomes more uniform:

$$\frac{p(a)}{p(b)} \downarrow 1 \qquad \text{or} \qquad p(a) \approx p(b)$$

- If temperature decreases ($\tau \downarrow$): distribution becomes more peaked (degenerate):

$$\frac{p(a)}{p(b)} \uparrow \infty \qquad \text{or} \qquad p(a) \gg p(b)$$

Optimistic vs. Realistic Initialization

How to ensure that every arm is sampled at least once (preferably: at beginning of exploration!)

- In optimistic initialization, the agent initializes its action-value estimates higher than the largest possible reward
- The agent always selects the greedy action
- Rewards are always disappointing, directing the agent to the un-explored arms

Quantifying Performance: Minimising Total Regret

- Cumulative reward
  - but compared to what?
- Choosing the optimal arm (Percentage)
  - Less indicative if two arms similar (near-optimal) rewards;
- Minimizing total regret:
  - Opportunity loss: what did we miss out on?
- How many trials before we are "pretty sure" we have identified the optimal arm? Confidence interval.
- Pure exploration:

Quantifying Performance: Minimising Total Regret

- **Optimal mean reward value** and **opportunity gap:** :

$$q^* = q(a^*) = \max_a q(a) \qquad \Delta_a := q^* - q(a)$$

- **Regret (opportunity loss) when taking action i:**

$$q^* - q(a_i)$$

- **Total regret** up to time T:

$$L_T = \sum_{t=1}^{T} \Delta_{A_i} = \sum_a \Delta_a N_T(a).$$

- Expected total regret (up to time T):

$$\ell_T := E(L_T) = \sum_a \Delta_a \, E N_T(a)$$

Expected total regret for ϵ-greedy

- Expected total regret:

$$\ell_t = \sum_a \Delta_a \, EN_t(a).$$

- For ϵ-greedy (when we have found optimal a^*):

$$EN_t(a) = \begin{cases} (1-\epsilon)t & \text{if } a = a^*, \Delta_a = 0 \\ \dfrac{\epsilon}{k-1} t & \text{if } a \neq a^*, , \Delta_a > 0 \end{cases}$$

Hence: $\ell_t = (\epsilon \overline{\Delta})\, t$ where $\overline{\Delta} = \dfrac{1}{k-1} \sum_{a \neq a^*} \Delta_a$

- For constant exploration (ϵ) total regret grows linearly in t,

Can we do better? Lai-Robbins

What if we reduce exploration ϵ over time?

Lai & Robbins (1985)

Asymptotically, total regret is at least logarithmic in number of steps:

$$\ell_t \geq A \log t \qquad \text{as } t \longrightarrow \infty$$

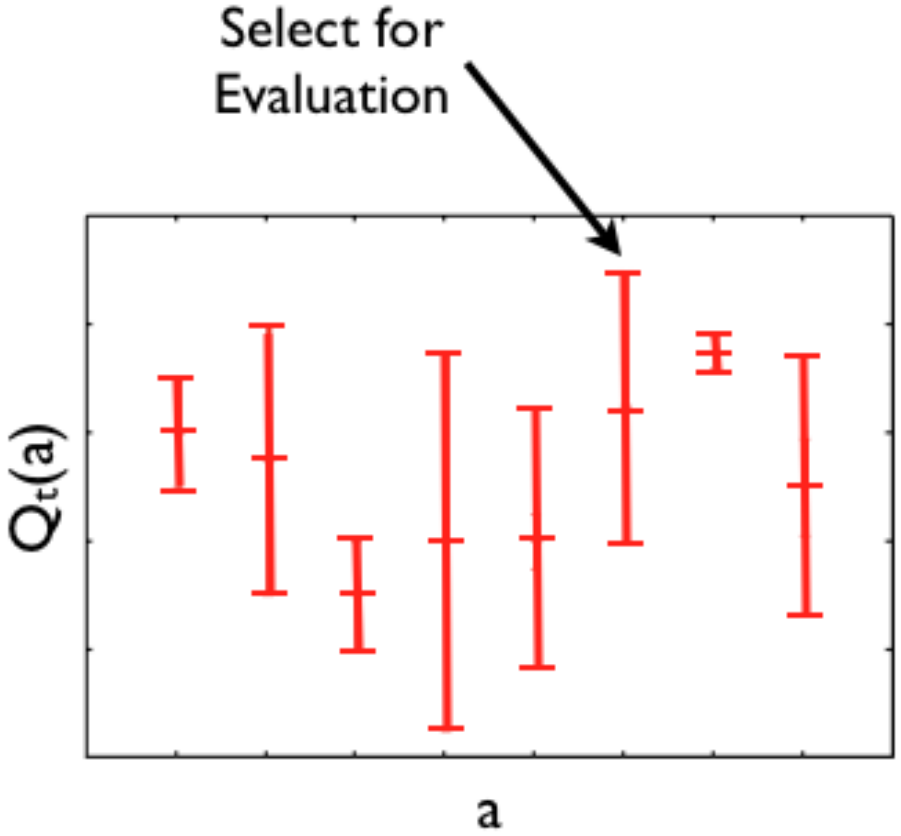
where

$$A = \sum_{a:\Delta_a > 0} \frac{\Delta_a}{KL(f_a; f_a^*)}$$

- $KL(f; g)$ Kullback-Leibler divergence;
- What exploration schemes give rise to this performance?

UCB: Upper confidence bounds

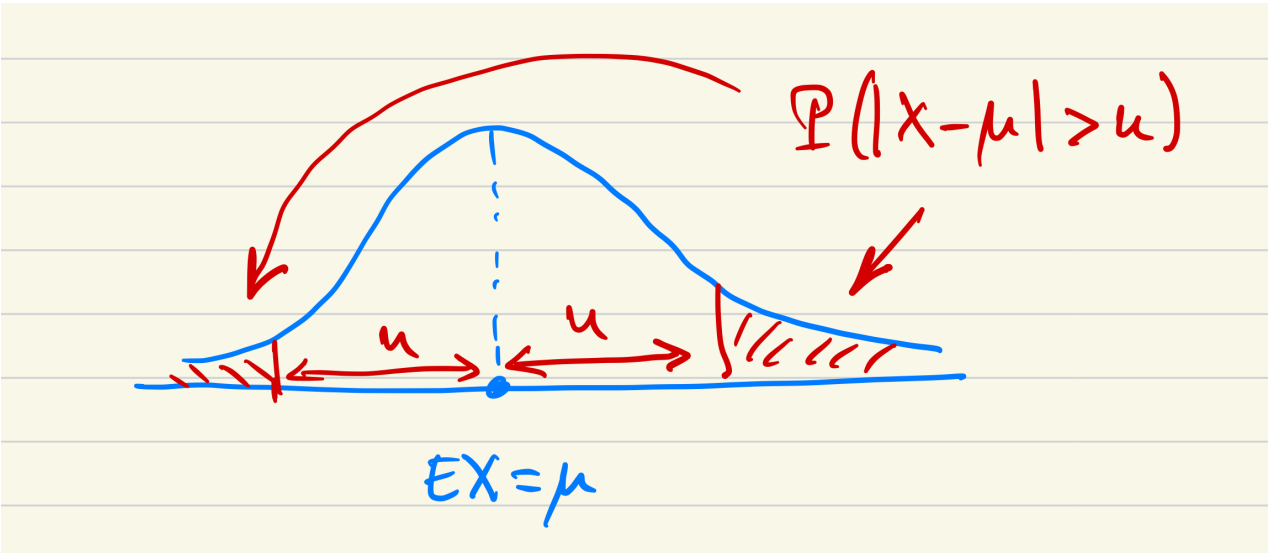
- **UCB: Optimism in the face of uncertainty!**
- Neither ϵ-greedy nor softmax consider uncertainty in action-value estimates
- Goal of exploration: reduce uncertainty
- So focus exploration on most uncertain actions
- Compute confidence intervals for each action
- Always take action with highest upper bound



Aside: Concentration Inequalities

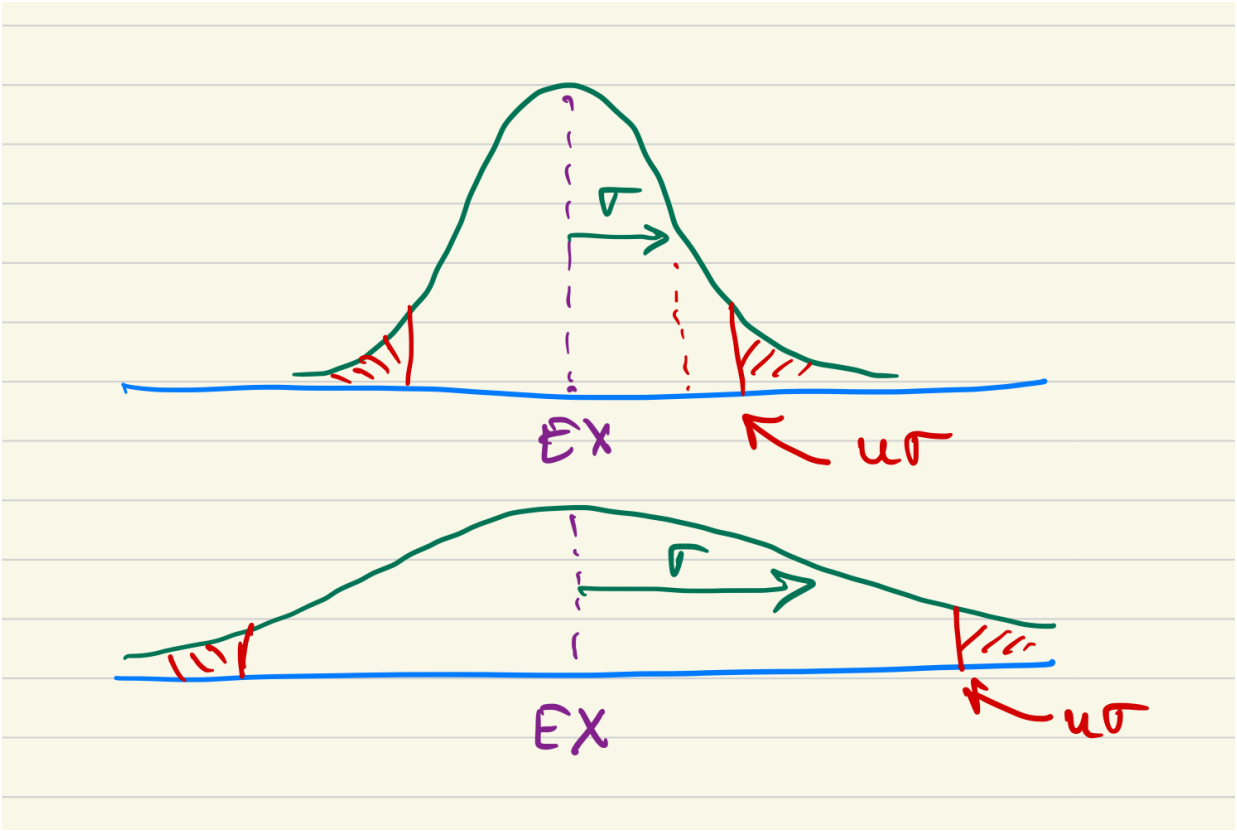
- **Concentration inequalities** provide bounds on how much a random variable X can deviate from some value (typically, its expected value $E(X)$):

$$P(|X - E(X)| > u) = \quad ??$$



Markov – Chebychev (3)

- Standard deviation σ provides natural yard stick (unit-length)!
- So it is natural to look at standardized deviation $|X - \mu|/\sigma$, i.e. deviation $|X - \mu|$ relative to σ.



Markov – Chebychev (3)

- For $Y \geq 0$ we have:

$$P(Y > u) \leq \frac{EY}{u}$$

- Now take

$$Y = \frac{|X - \mu|}{\sigma} \quad \text{then} \quad EY^2 = 1$$

$$P\left(\frac{|X - \mu|}{\sigma} > u\right) = P(Y > u) = P(Y^2 > u^2) \quad \leq \frac{1}{u^2}$$

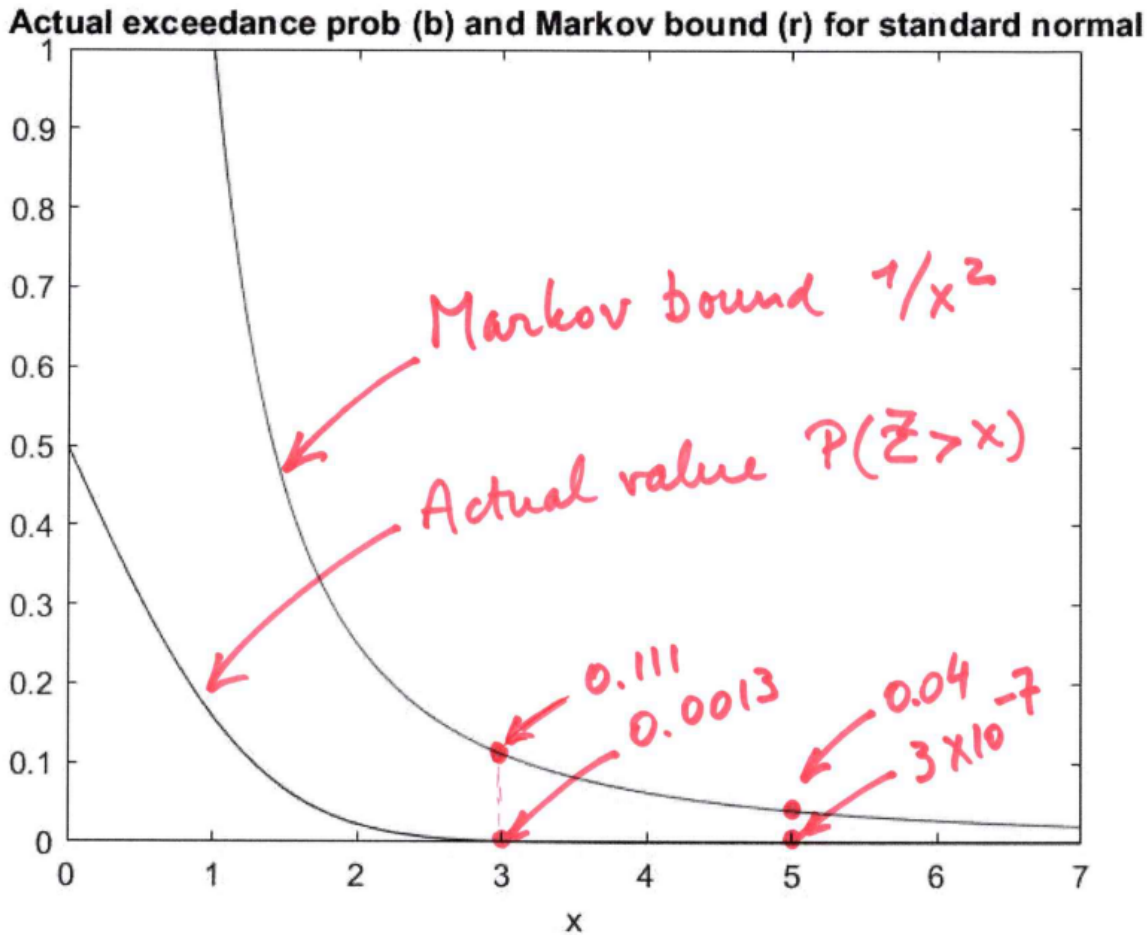
- Markov-Chebychev inequality:

$$P\left(\frac{|X - \mu|}{\sigma} > u\right) \leq \frac{EY^2}{u^2} = \frac{1}{u^2}$$

Markov – Chebychev (4)

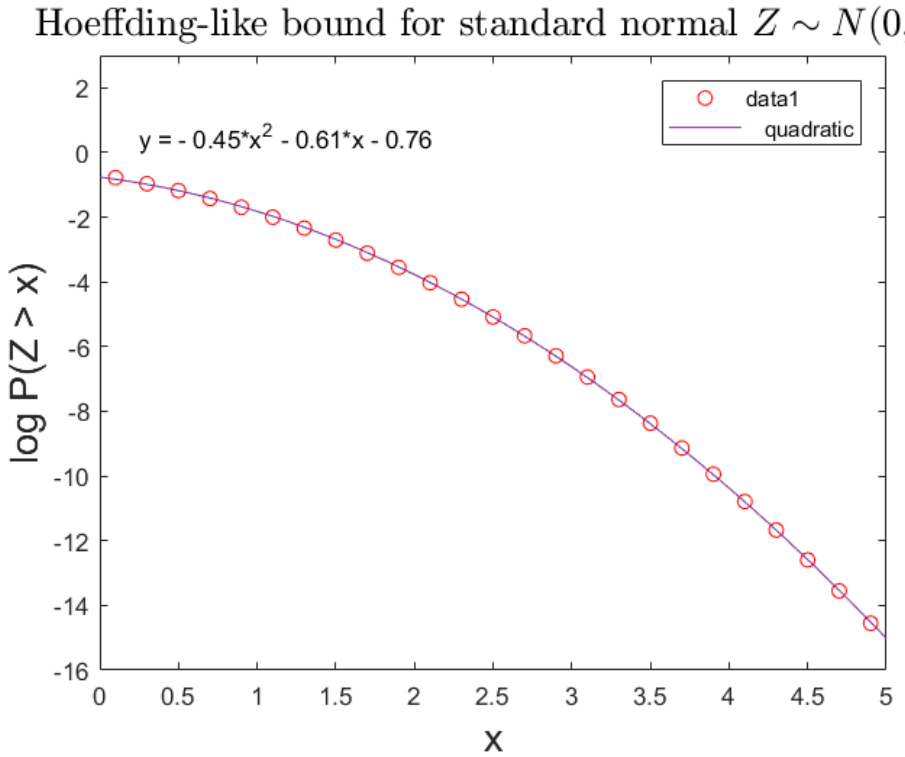
The Markov-Chebychev bound is not very tight!

$$\text{For } Z \sim N(0,1): \quad P(Z > x) \leq 1/x^2$$



Hoeffding inequality: Numerical experiment

From numerical experiments for $Z \sim N(0,1)$ we observe



$$\log P(Z > x) \approx -\alpha x^2$$

$$\log P(\sigma Z > \sigma x) \approx -\alpha x^2$$

$$\log P(\sigma Z > u) \approx -\alpha u^2/\sigma^2$$

$$P(\sigma Z > u) \approx e^{-\alpha u^2/\sigma^2}$$

$$P\left(\frac{\sigma}{\sqrt{n}} Z > u\right) \approx e^{-\alpha n u^2/\sigma^2}$$

Fron CLT: $\overline{X}_n \sim N(\mu, \sigma^2/n) \implies P(\overline{X}_n > \mu + u) \approx e^{-nu^2/2\sigma^2}$

Hoeffding inequality for sum of bounded rv's

Hoeffding's inequality

Let $X_1, X_2, \ldots, X_n$ be i.i.d. (bounded) random variables such that $c \leq X_i \leq d$ (where $L = d - c < \infty$) with mean $\mu := E(X_i)$.
Consider the sample mean:

$$\overline{X}_n = \frac{1}{n} \sum_{i=1}^{n} X_i$$

Then the exceedance probability is exponentially bounded:

$$P(\overline{X}_n \geq \mu + u) = P(\mu \leq \overline{X}_n - u) \leq e^{-2nu^2/L^2}$$

and similarly:

$$P(\overline{X} \leq \mu - u) \equiv P(\mu \geq \overline{X} + u) \leq e^{-2nu^2/L^2},$$

UCB: Applying Hoeffding to Upper Confidence Bounds

Hoeffding's inequality

$$P(\overline{X} \geq \mu + u) = P(\mu \leq \overline{X} - u) \leq e^{-2nu^2/L^2}$$

$$P(\overline{X} \leq \mu - u) = P(\mu \geq \overline{X} + u) \leq e^{-2nu^2/L^2},$$

Apply to estimation of bandit reward

How does **unknown real mean** $q(a)$ be estimated by **observed sample mean** $Q_t(a)$?

$$P\left(q(a) > Q_t(a) + U_t(a)\right) \leq \underbrace{e^{-2N_t(a)U_t(a)^2/L^2}}_{:=p(t)}$$

UCB: Applying Hoeffding to Upper Confidence Bounds

Hoeffding's upper bound:

$$P\left(q(a) > Q_t(a) + U_t(a)\right) \leq e^{-2N_t(a)U_t(a)^2/L^2} = p(t)$$

Given exceedance probability $p(t)$, solve for upper limit $U_t(a)$:

$$U_t(a) = c\sqrt{\frac{-\log p(t)}{N_t(a)}}, \qquad \text{where} \quad c = L/\sqrt{2.}$$

Now, let exceedance prob. go to zero over time: e.g. $\mathrm{p(t)} = t^{-1}$;

$$U_t(a) = c\sqrt{\frac{\log t}{N_t(a)}}.$$

UCB1-Algorithm: Upper confidence bounds Optimism in the face of uncertainty!

UCB1-Algorithm

$$a^*_{t+1} = \arg\max_a \left(Q_t(a) + c \sqrt{\frac{\log t}{N_t(a)}} \right)$$

- DON'T choose the arm that has performed best sofar, but...
- DO choose the one that could **reasonably** perform best in the future!
- Optimism in the face of uncertainty! :-)

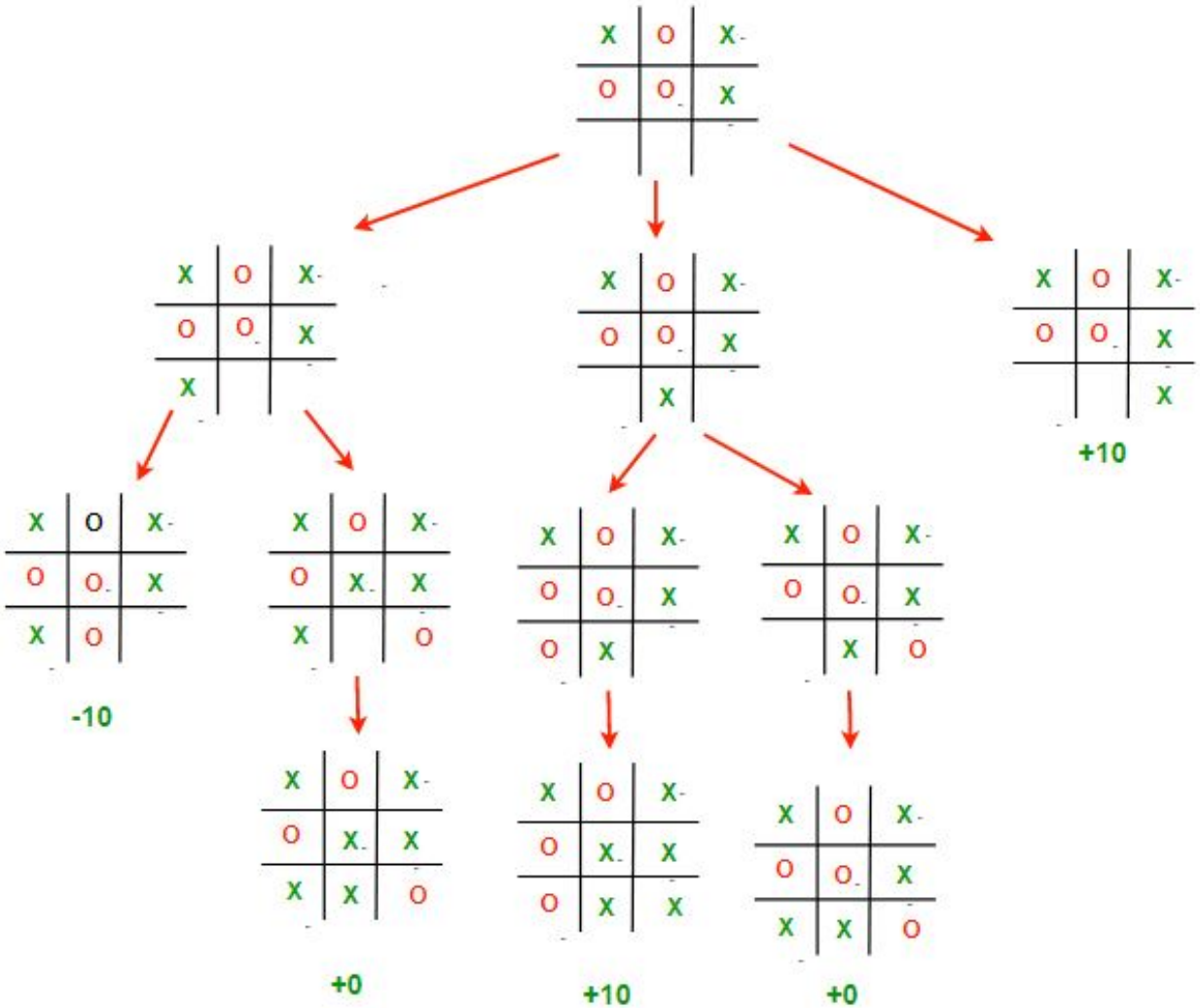
UCB1-Algorithm

$$a^*_{t+1} = \arg\max_a \left(Q_t(a) + c \sqrt{\frac{\log t}{N_t(a)}} \right)$$

- The parameter c is some tunable width parameter;
- Every time action a is sampled, $N_t(a)$ increases, hence $U_t(a)$ decreases;
- Every time a is sampled, the UCB for the *other* actions increases (since t increases); Every action is sampled eventually!
- UCB1 algo achieves logarithmic asymptotic total regret!

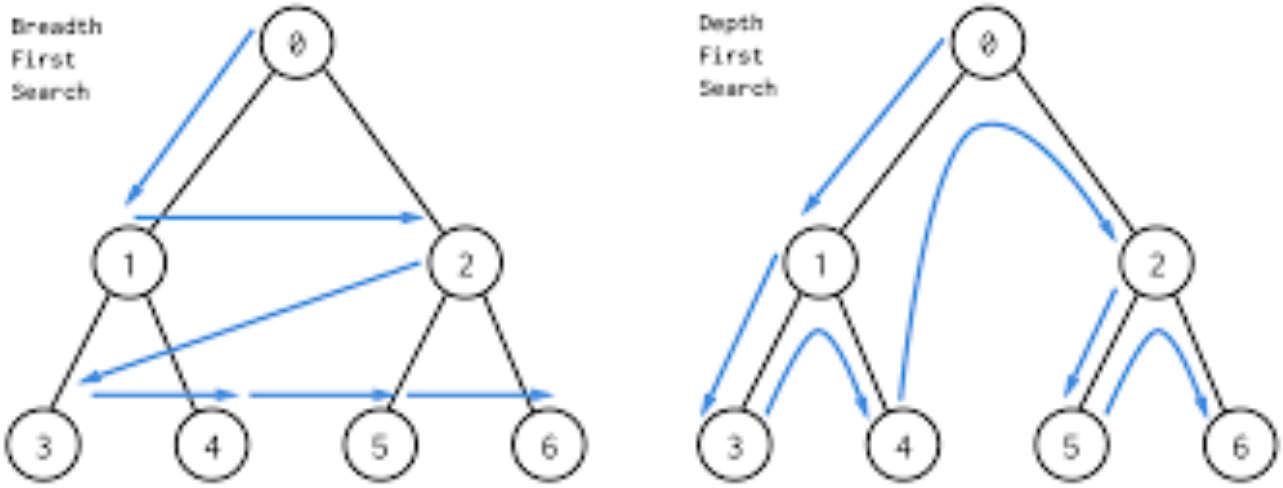
Tree Search

Tree search: generic problem underlying many AI tasks:



Tree Search Algo's

- Uninformed search: breadth or depth first search;



- A^* **search**: introduce heuristics that *inform search*;
- **Minimax search** (2 player adversarial) with $\alpha\beta$ pruning; Remove nodes outside optimal window;

Monte Carlo Tree Search (MCTS)

MCTS: main idea

- Decision-time planning
- Prioritise computational budget:
  - No systematic scan of all nodes
  - Primarily focus on nodes that look promising
- Balance exploration and exploitation, e.g.
  - ϵ-greedy, or UCB1 (UCT: Upper Confidence bounds for Trees)
- MCTS is an important ingredient in many of the recent success stories in game AI (e.g. AlphaZero GO);

MCTS: Amplification of loop

- **Tree traversal and node selection:**
  - use tree policy to construct path from root to (most) promising "tree policy (aka snowcap)" leaf node;
  - Tree policy: Always choose child node with best (but finite) UCB-value:

$$UCB(node_i) = \overline{x}_i + c\sqrt{\frac{\log N}{n_i}}$$

    $\overline{x}_i$: mean node value; n_i: #visits of node i; N #visits parent;
  - Proceed till tree policy leaf node has been reached: this node has children that haven't been explored. .
- **Expansion** of selected (tree policy) leaf node; i.e. Randomly pick unexplored child (action).
- **Simulation** based on roll-out policy
  - roll-out: *random* or *fast heuristic* (roll-out states not stored!);
- **Backup:** update values along tree traversal path ;

MCTS: Tree policy vs. roll-out

- Repeat loop (always starting in same root node) until predetermined computational budget has been exhausted.
- Then choose the best child-node as new root node and repeat.