1.0 Module Overview

What are Intelligent Agents (IAs):
- **autonomous decision making based on available information**
- IAs can interact with other IAs, forming a multi-agent system
- IAs have specific goals such as completing tasks, and **often compete for scarce resources** with other agents

IA vs Robotics/ML
- robotics focuses on hardware, while IA is more about the mind/software of the robot
- ML is more about extracting useful information from data, IA is more about making (optional) decisions based on available information

Why study IA:
- Finance/e-commerce
- Smart grid
- Autonomous vehicles
- Cloud computing
- Robotics
- IoT
- Kidney Exchange
- Security Patrolling
- Adversarial ML

Some deadlines:

Preliminary Deadlines
See module page for latest information

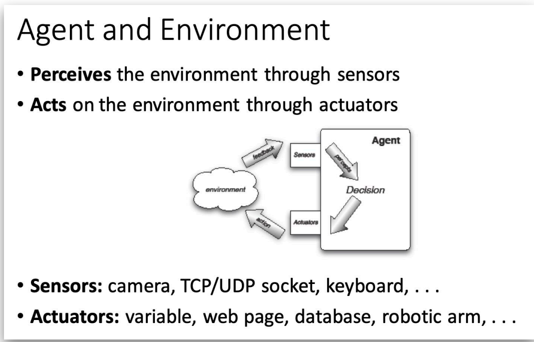
Teaching week	Day and Time	Deliverable
Week 3	Tuesday 18 October 4pm	Submit team information using this link
Week 4	Friday 28 October 09-11am	Assessment 1
Week 7	Friday 18 November 09-11am	Assessment 2
Week 9	Tuesday 29 November 4pm	Submit preliminary agent
Week 10	Friday 09 December 09-11am	Assessment 3
Week 11	Tuesday 13 December 4pm	Submit final agent (latest valid submission counts)
Week 15	Tuesday 10 January 4pm	Submit group report

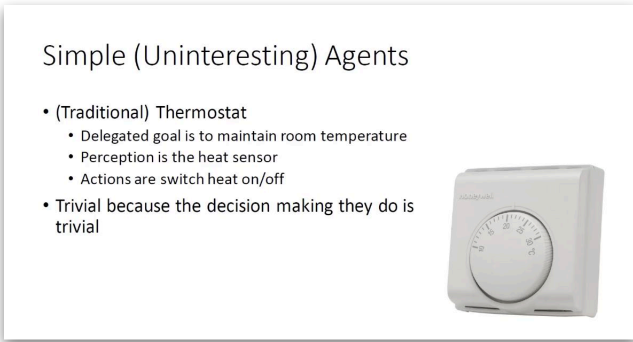
1.1 Introduction to Intelligent Agents

Five ongoing trends have marked the history of computing:
- **Ubiquity** - (the fact of appearing everywhere or of being very common) computational costs have lowered drastically, making powerful computational devices far more accessible to the general public
- **Interconnection** - the devices around us form a network of communication
- **Intelligence** - the complexity of tasks that we are capable of automating and delegating to computers has grown steadily
- **Delegation** - computers are doing more for us, without our intervention. We are giving control to computers (even in safety critical tasks). Eg fly-by-wire aircrafts, where the machine's judgement may be trusted more than an experienced pilot
- **Human-Orientation** - AI programs need input from humans (eg understanding preferences and for controlling and checking decisions). Explainable AI: need higher level of abstraction to interact with complex systems, using familiar interfaces. AI is not only about automation, but also augmenting human intelligence

How does this lead to the field of IAs and multi-agent systems:
- **Delegation** and **Intelligence** imply the need to build computer systems that can act effectively on our behalf
- This means that IAs should: **act independently**, **represent our best interest**, which implies that they need to be able **to cooperate and reach agreements (or even compete) with other systems** that have different interest (like we do with people)

What is an agent?
- "a computer system that is capable of <u>autonomous</u> action in some environment, in order to achieve its <u>delegated</u> goals"
- Key points are **Autonomy** (capable of independent action without need for constant intervention) and **Delegation** (acts on behalf of its user or owner)
- We think of an agent as being in a close-coupled, continual interaction with its environment (sense - decide - act - sense - decide …)





We typically think of intelligent agents as those exhibiting at least the 3 types of behaviour:
- Reactive
- Pro-active
- Social

Reactive:
- if the environment is fixed =>> an IA can execute actions blindly
- If the environment is dynamic (like in the real world) =>> IA must take into account the possibility of a failure - ask itself if it is worth acting! A reactive system is one that maintains an ongoing interaction with its environment, and responds to changes that occur in it (in time for the response to be useful) (i.e in real time)

Pro-activeness (generating and attempting to achieve goals, not driven solely by events; taking the initiative):
- reacting to an environment is easy (stimulus => response rules)
- But we ideally want agents to do things for us => so we need goal-directed behaviour
- Recognising opportunities

Social Ability:
- the real world is a multi-agent environment: being selfish and not cooperating with others could prevent you from fulfilling your goals
- **Some goals can only be achieved via cooperation**

Other properties:
- **Rationality**: agents will acts in order to maximise (expected) utils
- **Learning/adaptation**: agents improve performance over time

IA Challenges:
- How to represent **goals** and user **preferences** (Human-Orient challenge)
- How to optimise decision making
- How to learn and adapt and improve over time

Multi-Agent Systems (MAS):
- a MAS is a system that consists of a number of agents, which interact with each other
- In the most general case, agents will be acting on behalf of the users with different goals and motivations with one-another
- To successfully interact, they need to cooperate, coordinate and negotiate with each other as people do

MAS design challenges:
- how can cooperation emerge in societies of self-interested agents
- How can self-interested agents recognize conflict and how can they (nevertheless) reach agreement
- How can autonomous agents coordinate their activities so as to cooperatively achieve goals

Collaborative vs Self-Interested MAS:
- Agents in Collaborative MAS are owned by the same person/organisation. Agent design can be controlled and made to coordinate. These systems are more robust compared to “centralised” ones. Agents are also easy to control - no self-incentive is needed.
- Agents in Self-Interested/Competitive MAS work on behalf of someone else. The Agents can also be designed differently since there is no control over the design of other agents. Each agents has its own goals but ultimately the agents still need to cooperate. They need mechanism for resolving conflicts of interest.

Link with other disciplines

- The field of intelligent agents and multi-agent systems is highly interdisciplinary
  - Robotics
  - Complex Systems Science
  - Economics
  - Philosophy
  - Game Theory
  - Logic
  - Social Sciences
- This can be both a strength (infusing well-founded methodologies into the field) and a weakness (there are many different views as to what the field is about)

Classical AI and IA:

- Classical AI focuses on algorithms to solve hard decision problems (eg planning), interpreting complex data (ML and Machine Vision), knowledge representation and reasoning etc
- Agents often combine these techniques:
  - Robotics: vision + planning
  - Software assistants: ML + scheduling
  - E-commerce “bots”: search + ML

Some other views of the field

- Agents as a paradigm for software engineering:
  - Software engineers have derived a progressively better understanding of the characteristics of complexity in software. It is now widely recognised that interaction is probably the most important single characteristic of complex software.
- Agents as a tool for understanding human societies:
  - Multi-agent systems provide a novel new tool for simulating societies, which may help shed some light on various kinds of social processes.

2.1 Agent-Based Negotiation - Informally

Conflict of interest between agents:
- arises when agents have different preferences or goals
- Something people do on a daily basis
- In MAS, conflicts arise **when agents are self-interested** (**maximising their own benefit/utility function**) AND **the agents represent different stakeholders** (different sets of preferences/goals)
- **Example conflict:** a buyer wants to pay as little as possible, while a seller wants to maximise profits

Conflict Resolution:
- **possible if there is a mutual benefit to reach an agreement** (in other words, it is possible if reaching an agreement is better than continuing the conflict)
- Conducted via **Auctions**, **Voting** and **Negotiation**
- The resolution approach should be chosen based on the context of the problem

Auctions:
- **Primary used to allocate scarce resources or tasks**. For example, items to buyers, ad space for advertisers, tasks to robots, stocks and shares etc
- *Requirement 1)* **Needs a clearly defined protocol** (rules or mechanism)
- *Requirement 2)* **Requires a trusted third party** (auctioneer)
- *Requirement 3)* Often involves a **continuous resource** such as money
- *Requirement 4)* Exploits competition between agents (works better with more agents)

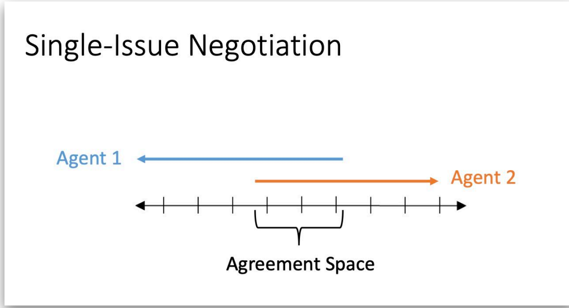
Voting:
- used for group based decisions (aka *social choice*):
- *Requirement 1)* **A single decision from a (typically finite) number of options**
- *Requirement 2)* Each agent can have different preferences for each option (**given by a preference order - ordinal utility function**)
- *Requirement 3)* **Clearly defined protocol** that governs how/howManyTimes/who can vote

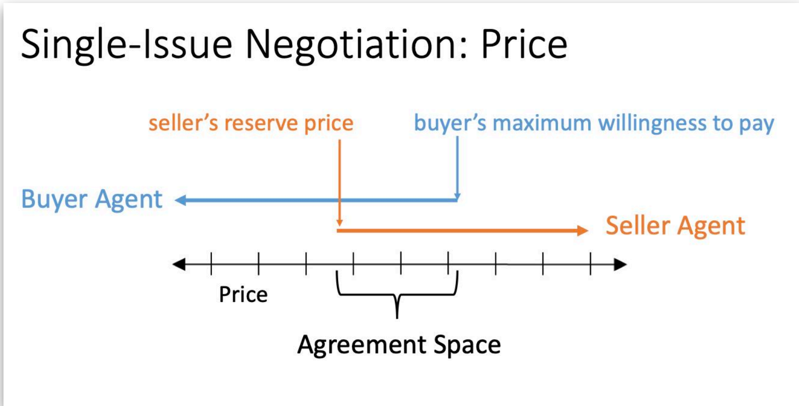
Negotiation (aka Bargaining):
- governed by a **protocol** which defines the “rules of encounter” between the agents, including - type of communication & proposals that are allowed as well as who can make what protocol at what time
- **More flexible than other approaches**:
  - Protocol typically involves **exchanging offers**, but can include other info such as **arguments** (reasons why)
  - Allows for less structured protocols
  - Often bilateral (between two agents) but some protocols allow multi-party negotiation
- Enables more complex types of agreements (eg multi-issue negotiation)
- **Often decentralised** but can involve a third-party mediator (agent-mediated negotiation)
- **Negotiation environment** (“what are we actually negotiating about?” == set of possible outcomes/agreements):
  - Single-issue negotiation (aka distributive bargaining): eg. Negotiating about a price. This is a win-lose negotiation, which encourages competitive setting
  - Multi-agent negotiation (aka integrative bargaining): more complex negotiation which includes more issues and allows for mutual benefit and cooperation

- **Agent preferences** (the preferences over all possible agreements (and disagreements), typically specified using **utility functions**)
- **Agent negotiation strategies (specifies the behaviour of the agents):**
  - Actions employed (e.g. the proposals made) at each possible decision point, given the information available to the agent

2.2 Agent-Based Negotiation - Negotiation Protocols

Most common Negotiation protocols:
- Ultimatum Game
- Alternating Offers
- Monotonic Concession Protocol
- Divide and Choose





The Ultimatum Game (aka "take it or leave it" game) - Bilateral Negotiation:
- agent 1 suggests a division of the cake
- Agent 2 can only choose to accept or reject the cake
- If agent 2 rejects, no-one gets any cake (the cake gets thrown away. The cake represents the utility gain that each agent can earn if they resolved their conflict)

Discussion about the Ultimatum Game:
What would good strategies be of the agents if they are completely selfish (and rational)? Suppose agent 1 offers agent 2 half the cake. Will agent 2 accept if they are rational?. What about less than half the cake? What about a tiny silver of cake. Assuming Agent 2 is rational, it should accept any offer as long as they gain something (Remember the cake gets thrown away if we dont reach an agreement). This type of thinking is formalised in Game Theory which will be covered later.

Agent 1 has a clear advantage, the so-called first-mover advantage. **This results in outcomes that can be seen as unfair.** The Ultimatum game **assumes** that the "agreement space" (aka the size of the pie) is known. I.e the seller's reserve and buyer's willingness to pay are shared knowledge. In practise this assumption is not always true.

If the pie is unknown, the Ultimatum Game does not support the ability to explore the negotiation space, so we often can't reach an agreement. This is even worse when there are multiple issues (as we will discuss later).

Alternating Offers Protocol:
- negotiation consists of a number of rounds
- Agents exchange offers in an alternating fashion
- First player starts with proposing an offer. Then the second player can either accept or reject and counter offer, or reject and break-off negotiation.
- Negotiation usually ends after set number of rounds (a deadline) or if either player breaks off the negotiation

Discussion:
Although the protocol offers exchange of more rounds, is there any incentive for any of the agents to concede? **Note that there is a final round which is crucial here. <u>The last round is identical to the Ultimatum Game.</u> So depending on who gets to make the last offer, they can again propose a take-it-or-leave it offer, which a rational agent would accept**. So in a way we haven't really resolved the problem with the Ultimatum Game => we need some way to incentivise a connection. This can be due to some form of time pressure or by changing the protocol.

Monotonic Concession Protocol:
- Again, negotiations proceed in rounds
- In each round, **the agents simultaneously propose offers (without seeing the other offer)**
- If both offers "match" (eg buyer offers a price that is more than the seller is offering) then one of them is chosen
- If offers don't match, when negotiation proceeds to the next round
- **In the next round at least one of the agents need to concede (by concede we mean alter their original offer (concede some of the utility gain that they will earn compared to the previous round))**
- **If neither of the agent concedes, the negotiation ends without a deal**

Divide and Choose:
- in case the "cake" denotes **a continuous resource that needs to be divided between agents** (kinda like what we use Auctions for?? == resource allocation)(eg inheritance money or land) then the "divide and choose" protocol can be used. Note it does not make sense for other types of negotiations such as price negotiation. The protocol consists of 2 steps:
  - First, agent 1 divides the cake into 2 portions
  - Then, agent 2 chooses on of the portions

Since agent 2 is going to choose the best portion for them, agent 1 has an incentive to divide the cake equally. So each agent ends up with half. Remember that this protocol only works for so-called resource allocation problems, where there is some (continuous) resource that needs to be divided between multiple agents. This protocol also works for **non-homogeneous resources (eg when parts of the cake are more attractive, in the case of land division)**. There are equivalent protocols for more than 2 agents, but this gets more complicated.

Three desirable properties of a negotiation:
- **a deal should be better than no deal for all agents (individual rationality)**
- There should be "no money" (or cake) left on the table **(Pareto efficiency). Everyone can earn more utility gains without hurting the other's rewards.**
- The agreement should be "**fair**"

2.3 Agent-Based Negotiation - Formally

We model an agent's preferences over outcomes using a **utility function** $U(o)$ where:

- $o \in O$ is the offer
- O is the set of possible offers

Two types of utility functions:

- Ordinal Preferences
- Cardinal Preferences

Ordinal Preferences:

- a preference order over outcomes is specified but there is no numerical utility
- Eg. $U(o_1) > U(o_2)$ means that o_1 is preferred over o_2
- The agent specify what they want but not by how much

Cardinal Preferences:

- $U(o_1) = 0.78$, $U(o_2) = 0.5$
- We can infer ordinal preferences from cardinal ones

Example: Buyer/Seller Price Negotiation

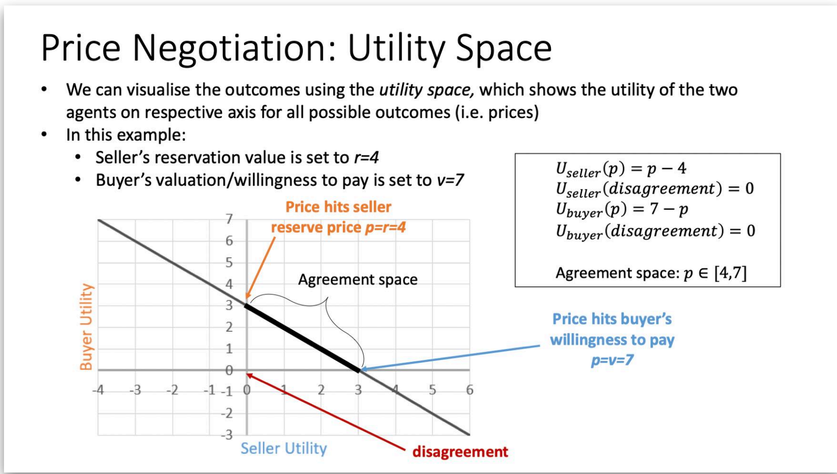
- set of possible outcomes O are: **{the price p, disagreement}**
- Cardinal utility functions are given by

$$U_{seller}(p) = p - r$$
$$U_{seller}(disagreement) = 0$$
$$U_{buyer}(p) = v - p$$
$$U_{buyer}(disagreement) = 0$$

Where

r is the seller's opportunity cost/reserve value (= minimum price that they are willing to sell)
v is the maximum willingness of the buyer to pay (buyer's value for the good)

Seller wants to maximise p so that they maximise $U_{seller}(p) = p - r$
Buyer wants to minimise p so that they maximise $U_{buyer}(p) = v - p$



Negotiation cannot go on for infinity so we need to introduce “**Time Pressure**” into the negotiations. There are different ways to model them:

- **Deadlines** = **imposed by protocol** (eg the Ultimatum game is 1 round). They can also be determined by individual constraints (in which case deadlines of agents can differ)
- **Break-off probability** = imposed by protocol (Eg like the monotonic concession protocol). Alternatively, agents can decide to break-off negotiations themselves (like in the alternating offers protocol)
- **Bargaining costs** (costs associated with each round of the negotiation -> so we penalise long negotiations)

Modelling Bargaining/Negotiation costs:

- **Fixed Costs**. Let c_i be the costs for agent i and t is the time or bargaining round. Then the utility at time t is given by $U_i^t = U_i - t \cdot c_i$. Note that c_i is fixed so it is the same for every round of the negotiation
- **Discount Factors**: an analogy is that of a “Melting” ice cake - the longer you wait, the more the cake “shrinks”. Let $\delta_i < 1$ be the discount factor of agent i (different agents can have different discount factors), then the utility is given by $U_i^t = U_i \cdot \delta_i^t$. Note that δ_i^t is δ_i to the power of t. So the bigger t gets, the bigger the penalty becomes.

Multi-Issue Negotiation:

- negotiations often involve multiple other issues, such as time, quality of service, delivery time etc
- In that case an offer/outcome o is a vector consisting of a value o_j for each issue j
- We often use a **weighted additive utility function**:

$U_i(o) = \sum_j w_{i,j} \cdot U_{i,j}(o_j)$ where $U_{i,j}(o_j)$ is the utility for issue j

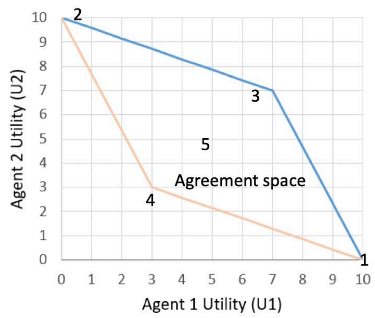
For example:

- suppose there are two cakes A and B and offer o_i represents the share that agent 1 receives for cake $i \in \{A, B\}$ (so agent 2 gets $1 - o_i$)
- So we have $U_{1,i}(o_i) = o_i$ and $U_{2,i}(o_i) = 1 - o_i$ for $i \in \{A, B\}$
- Agent 1 prefers cake A and agent 2 prefers cake B. In particular, agent 1 and agent 2 have weights 7,3 and 3,7 respectively.
- This results in utility functions:
- $U_1(o) = 7 * o_1 + 3 * o_2$
- $U_2(o) = 3 * (1 - o_1) + 7 * (1 - o_2)$

Example Utility Space (Answer)

$U_1(o) = 7 * o_1 + 3 * o_2$
$U_2(o) = 3 * (1 - o_1) + 7 * (1 - o_2)$

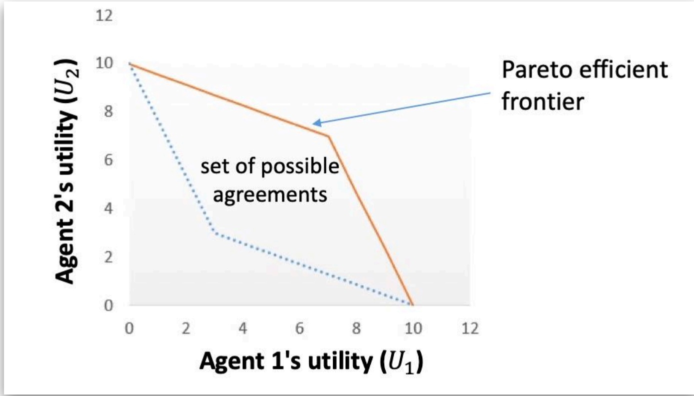
Offer	o_1	o_2	U_1	U_2
1	1	1	10	0
2	0	0	0	10
3	1	0	7	7
4	0	1	3	3
5	0.5	0.5	5	5



We can explore a lot more offers, and at some point we will start seeing that the utility scores are bounded in an “agreement space”

Pareto-Efficient Agreements:

- an agreement is said to be Pareto efficient (or Pareto optimal) if no further improvement is possible in the utility of one agent, without reducing the utility of the other agents
- "There is no agreement where both agents will be better off"
- **Pareto efficient frontier: set of all possible Pareto efficient agreements**
- Taking previous example, offer 3 is the best one
- The Pareto efficient frontier (orange line) is the set of all possible Pareto efficient agreements



Again Desirable Properties:

- agreements should be **individually rational**: $U(o) > U(disagreement)$
- Agreements should be **Pareto efficient**
- Agreements should be **fair**

2.4 Agent-Based Negotiation - Fairness

There are many different concepts of fairness. The four main ones are:
- **Utilitarian Social Welfare:** maximise the sum of utilities (aka social welfare). **Formally**: $max_{o \in O} U_1(o) + U_2(o)$
- **Egalitarian Social Welfare:** maximise the minimum utility. Formally $max_{o \in O} min(U_1(o), U_2(o))$
- **Nash Bargaining Solution**: maximise the **product** of the utility of the agents (minus the disagreement payoff).
  $max_{o \in O}(U_1(o) - U_1(disagreement)) \cdot (U_2(o) - U_2(disagreement))$
- **Envy-freeness**: no agent prefers the resources allocated to other agents (only makes sense in resource allocation settings)

Nash Bargaining is **uniquely characterised** by the following **axioms/properties**:
- **Individual Rationality** (the solution always satisfies $U_1 \geq U_1(dis)$ and $U_2 \geq U_2(dis)$
- **Pareto Efficiency**
- **Invariance to equivalent utility representations** (the solution is insensitive to *affine transformations*) What does that mean? Suppose I multiple my utility function by 5 but the other agent's do not. While these could suggest that my utility is more important. But in fact what does utility even mean? What does that value represent? It doesn't necessarily have any meaning - it is relative to your own utility function. You can't just compare the utility scores of agents because their utility functions could be different/use different units etc. So an affine transformation is for example if I multiply a utility function by 5.. but since we are maximising the product of the utility functions, we end up with exactly the same solution, no matter how we multiply our utility function.
- **Independence of irrelevant alternatives (IIA)**: remove all the non-optimal agreements, and the optimal agreements remain the same
- **Symmetry (SYM)**: if the agents have the same preference, then the solution gives them the same utilities

Envy-Freeness:
- an agent is envious of the other agent, if it would prefer the allocation received by that agent. A solution is envy free if no agent prefers the allocation of another agent (no agent is envious)
- This concept only makes sense in resource allocation tasks
- Example: suppose we have two pies (2 issues) and 2 agents, and o_j is the share of pie j that goes to agent 1, and $1 - o_j$ goes to agent 2. Then agent 1 is envious if: $U_1(1 - o_1, 1 - o_2) > U_1(o_1, o_2)$. Similarly Agent 2 is envious if: $U_2(o_1, o_2) > U_2(1 - o_1, 1 - o_2)$

Exercise: Example 1 Revisited

Considering the example from before, which offer(s) satisfy:

- Utilitarian social welfare solution
- Egalitarian social welfare solution
- Nash Bargaining solution
- Envy-free

Example 1 Revisited

$U_1(o) = 7 * o_1 + 3 * o_2$
$U_2(o) = 3 * (1 - o_1) + 7 * (1 - o_2)$

Try and fill out the table below yourself and to come up with the answers.

Offer	o_1	o_2	U_1	U_2	Utilitarian Sum	Egalitarian Min	NBS Product
1	1	1	10	0			
2	0	0	0	10			
3	1	0	7	7			
4	0	1	3	3			
5	0.5	0.5	5	5			

Example 1 Revisited: Envy Free

$U_1(o) = 7 * o_1 + 3 * o_2$
$U_2(o) = 3 * (1 - o_1) + 7 * (1 - o_2)$

Compute agent utilities when their allocation is REVERSED (i.e. Agent 2 is getting what Agent 1 is getting, and vice versa)

Offer	o_1	o_2	U_1	U_2	U'_1	U'_2	Envious?
1	1	1	10	0			
2	0	0	0	10			
3	1	0	7	7			
4	0	1	3	3			
5	0.5	0.5	5	5			

Example 1 Revisited: Envy Free

$U_1(o) = 7 * o_1 + 3 * o_2$
$U_2(o) = 3 * (1 - o_1) + 7 * (1 - o_2)$

Compute agent utilities when their allocation is REVERSED (i.e. Agent 2 is getting what Agent 1 is getting, and vice versa)

Offer	o_1	o_2	U_1	U_2	U'_1	U'_2	Envious?
1	1	1	10	0	0	10	Agent 2
2	0	0	0	10	10	0	Agent 1
3	1	0	7	7	3	3	No
4	0	1	3	3	7	7	Agent 1 & Agent 2
5	0.5	0.5	5	5	5	5	No

There are many others. E.g. (1.0,0.1) is envy free as well, and even (1.0,0.2). However, (1.0,0.3) is no longer envy free, since at that point agent 2 prefers agent 1's allocation. CHECK YOURSELF!

Example 1 Revisited (Answers)

$U_1(o) = 7 * o_1 + 3 * o_2$
$U_2(o) = 3 * (1 - o_1) + 7 * (1 - o_2)$

Offer	o_1	o_2	U_1	U_2	Utilitarian Sum	Egalitarian Min	NBS Product
1	1	1	10	0	10	0	0
2	0	0	0	10	10	0	0
3	1	0	7	7	14	7	49
4	0	1	3	3	6	3	9
5	0.5	0.5	5	5	10	5	25

2.5 Agent-Based Negotiation - Negotiation Strategies

Game theoretic:
- Tries to formally analyse it from a mathematical perspective
- assumes rules of the game, preferences & believes of all players are common knowledge (**the pie**) (often unrealistic assumption, there is usually a lot of uncertainty for the other agent's utility functions)
- Assumes full rationality on the part of all players (=unlimited computation) (often this assumption does not hold true in more complex games/scenarios)
- Preferences encoded in a limited set of player types
- Closed systems, predetermined interaction, small sized games
- **Nash equilibrium** (discussed later)

Heuristic perspective:
- no common knowledge or perfect rationality assumptions needed
- More flexible
- Agent behaviour is modelled directly
- Suitable for open, dynamic environments
- Space of possibilities is very large

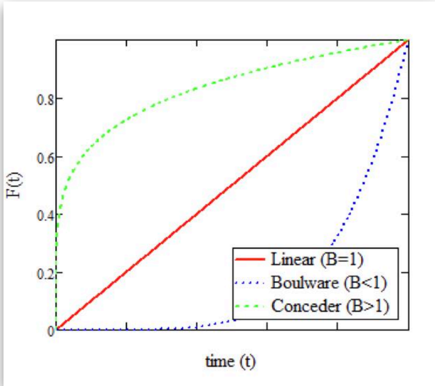
Heuristics are often used when there is some unknowns about the opponent (Eg utility function or the strategy they are using)

A strategy is made up of two parts:
- **Concession Strategy** (when we negotiate, we need to decide how much of our utility are we willing to sacrifice in order to make a good deal. Obviously, we want to maximise our own utility (that is always the goal), but if we start the negotiations using the offer that is the most attractive to us, we might not get an agreement if we stick to it. Therefore we need a **concession strategy** == "how quickly are we going to concede?", what should be the target utility I think I should achieve at a particular point in the negotiation)
- **Multi-issue offer producing strategy**: once a target utility is established, what offer to produce at or around that target utility (to ensure Pareto efficiency). This is trivial in case of single issues.

Well-known **Concession Strategies** include:
- **Time-Dependent tactics** (only depend on time/current round of the negotiation and not on the opponent's action). So Concession here is a function of time only.
- **Tit-for-tat** (relies on the behaviour of the opponent)

Time-dependent tactics:
- Let U_{max} and U_{min} denote the agent's max & mi acceptable reserve price respectively. Target utility offered at time t will be: $U_{target}(t) = U_{min} + (1 - F(t))(U_{max} - U_{min})$ where $F(t)$ is a function between 0 and 1 and gives the fraction of the distance between best and worst offer (eg using a function such as $F(t) = (\frac{min(t, T_{max})}{T_{max}})^{\frac{1}{\beta}}$ where T_{max} is the deadline and β is a constant
- Note that $F(0) = 0$ and $F(T_{max}) = 1$

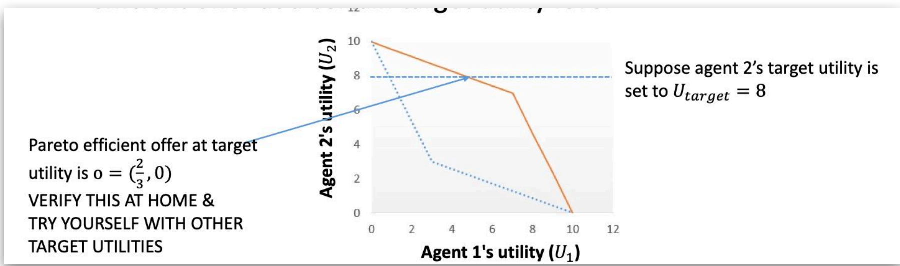
- **Hard-headed** ($\beta \to 0$): No concession. Sticks to the initial offer throughout (hoping the opponent will concede)
- **Linear time-dependent concession** ($\beta = 1$). Concession is linear in the time remaining until the deadline.
- **Boulware** ($\beta < 1$): Concedes very slowly; initial offer is maintained until just before the deadline. Probably most common.
- **Conceder** ($\beta > 1$): concedes to the reservation value very quickly. Useful if under time pressure (eg discount factor is very big) because we can concede faster which results in quicker agreements



Tit-for-tat concession strategy:
- “If the opponent is not conceding, why should I concede?”. With this strategy, we concede only if the opponent is also conceding.
- the agent detects the concession the opponent makes during the previous negotiation round, in terms of increase in the agent’s own utility function (Why? Suppose the opponent is conceding, but because they do not know your utility function, then maybe the offer they proposed is not useful to you. So we should determine how much the opponent is conceding using our own utility function and not theirs).
- The concession the agent makes in the next round is equal to (or less than) the concession made by the opponent in the previous round
  $concession \leq U_{own}(o^{t}_{opponent}) - U_{own}(o^{t-1}_{opponent})$
- As long as offer falls in the acceptable region (eg for price negotiation, above the reservation price of the seller and below that of the buyer)

Optimal Concession Strategy:
- if everything is known, then it is easy to compute a **best response**, **i.e the optimal concession given the utility and strategy of the opponent**.
- Note that the optimal strategy depends on the opponent but also the time pressure (Eg time discounting or deadline)
- However, often, there is a lot of uncertainty about the opponent’s utility function and strategy. In that case, machine learning techniques can be used to try and **learn the opponent model.** Learning the opponent model is not just about the strategy, but also about deriving what their utility function is.
- Also opponent might be doing the same, Therefore, game-theoretic approaches (where you reason about the opponent) can be useful

Offer-Producing strategy:
- **so far, the strategies discussed only provide a target utility**. In a multi-issue negotiation, you need to generate a value for each issue.
- It is a good idea to ensure that the offer is always Pareto efficient (otherwise we will be leaving money on the table!)
- We can only plot the agreement space if we know the utility functions of the two agents. So it is very important that we try to estimate it (eg using a model)
- If the utility functions are known, it is possible to calculate the Pareto-efficient offer at a certain target utility level



Unknown Opponent Utility:
- typically he opponent utility is not known, ie this is so-called *private information*
- Possible to guess the opponent utility function based on the offers received so far and the concessions observed
- The opponent **is most likely to concede on their least preferred issues first** (in case of additive utility functions in multi-issue negotiations) (eg two cakes, but agent 1 hates the first cake so they are likely to concede on it), and so you can use this to guess the weight of that issue

- Many approaches in the literature and this will be discussed in more details in the coursework labs

Preference Uncertainty and Elicitation:
- In many cases even "own" utility functions is not (fully) known by the agent (who is negotiating on behalf of a human). The agent has to ask the user lots of questions about their preferences. We can't just ask "what are your weights for issue 1, issue 2 etc". That is not how it works in reality. We also don't want to burden the user with too many questions (this is known as a "cognitive cost")
- Utility function obtained through a process called ***preference elicitation*** (Has associated "**cognitive costs**"; **trade-off between minimising cognitive cost and maximising utility**)
- Some recent work has looked at negotiation with *preference uncertainty*: how should we negotiate when we have incomplete information about own utility
- Preference uncertainty and elicitation will be explored as part of the coursework and more details will be discussed i a future lecture and the labs

3.1 Introduction to Utility Theory

A game is a mathematical model of interactive decision making. Different agents/players interact with the goal to achieve the best possible outcome. “Best outcome” might mean different things: winning, maximising payoff, etc. In order to understand what this means, we need to define a formal model of decision making.

Basic setting:
- for an agent, making a decision means selecting one among multiple choices
- Let $S = \{s_1, \ldots, s_m\}$ denote the choices available to an agent
- Making a choice leads to a certain outcome
- Let $\Omega = \{w_1, \ldots, w_2\}$ denote a set of outcomes
- Ω may be = all the possible outcomes of game of chess / the possible outcomes of negotiations between nations / the possible outcomes of an eBay auctions etc
- In these lectures we always assume the sets S, Ω to be **finite**
- An outcome function $g : S \to \Omega$ maps each choice to an outcome
- g **specifies the consequence of making a certain choice**
- How does an agent select the best outcome? We assume that, given any pair of outcomes (w, w'), an agent can say which of them they prefer.

Preferences:
We use slightly different interpretations of preferences, depending on whether the decision-making setting is one of certainty or uncertainty:
- in decision-making under **certainty**, we know exactly what the consequences of our choices will be
- In decision-making under **uncertainty**, we don’t know exactly what the consequences of our choices will be. For every possible choice, there are multiple possible consequences, each with an attached probability (more on this later)

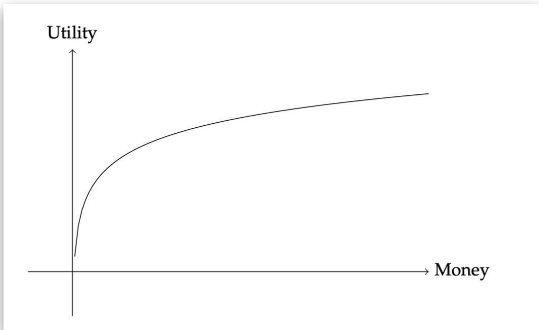
Formally, a **preference** is a binary relation $\succeq$ on $\Omega \times \Omega$. We assume this relation is:
- **Reflexive**: $w \succeq w$ for all $w \in \Omega$ “we prefer an outcome as much as the outcome itself”
- **Total**: for all $w, w' \in \Omega$, either $w \succeq w'$ or $w' \succeq w$
- **Transitive**: for all $w, w', w'' \in \Omega$, if $w \succeq w'$ and $w' \succeq w''$, then $w \succeq w''$
- Given a pair of outcomes (w, w'), $w \succeq w'$ means that an agent **prefers w at least as much as w'**

If both $w \succeq w'$ and $w' \succeq w$ then we are **indifferent** between the two => $w \sim w'$

If both $w \succeq w'$ but not $w' \succeq w$, then we **strictly prefer** w over w' => $w \succ w'$

Utility functions:
- to build a mathematical model we define functions **to model preferences**
- A **utility function** is a mapping $u : \Omega \rightarrow R$
- A utility function u over Ω is said to **represent** a preference relation $\succeq$ iff, for all $w, w' \in \Omega$: $w \succeq w'$ iff $u(w) \geq u(w')$
- Given a finite set Ω, for every preference relation $\succeq$ on $\Omega \times \Omega$ there exists a utility function $u : \Omega \rightarrow R$ that represents it. This is true for the finite case. In the infinite case, this is a bit more complex.
- **Utility functions are a numerical representation of preferences**
- An agent chooses an outcome over another **not because the numerical value of the utility, but because of their preference (the notion of preference <u>comes before</u> the notion of utility. An agent chooses an outcome that has a higher numerical value because that's the utility/numerical representation of their preference)**
- An agent choosing the best outcome is maximising their utility, as long as the representation is the correct one
- The numerical values are useful for adopting numerical techniques for the mathematical models, but the values themselves are not important. **Their relative order is. It is how agents rank their preferences.**
- **Utilities of different agents cannot in general be compared.**

Utility values vs Money:
- Utility values are not money. They can sometimes be interpreted as such, but they measure different things. Numerical **representation of individual preferences** vs objective numerical scale. Money rely on an objective numerical scale, whereas utility values are personal.



We can formally represent decision-making as a tuple $(S, \Omega, g, \succeq, u)$, where:
- S and Ω are the set of choices and consequences/outcomes
- $g : S \rightarrow \Omega$ is an outcome function that specifies the consequences of each choice
- $\succeq \subseteq \Omega \times \Omega$ is the agent's preference relation
- $u : \Omega \rightarrow R$ is the utility function representing $\succeq$
- A rational decision maker is one that makes the best possible choice
- This means that the agents make the choice that maximises their utility: $s \in argmax_{s \in S}\ u(g(s))$
- It is worth noticing that using numerical utilities allows us to express decision problems as optimisation problems

Decision under uncertainty:
- in many settings, we don't know exactly what outcome will result by making a choice
- For every options $s \in S$, there will typically be a range of possible outcomes, with differing probabilities of occurring
- Such settings require more complex machinery for preferences and utilities
- In particular, preference relations $\succeq$ over outcomes are not enough: we need preferences over **lotteries**

To represent the uncertainty of an outcome we use probability distributions:

- a probability distribution over a non-empty set of outcomes Ω is a function $\mu : \Omega \to [0,1]$ such that $\sum_{w \in \Omega} \mu(w) = 1$
- A probability distribution represent the likelihood of the occurrence of a certain outcome

Lotteries:

- a lottery over a set of outcomes Ω is a **probability distribution over Ω outcomes**
- We denote a lottery by L
- Example: suppose $\Omega = \{chocolate, vanilla, strawberry\}$ and μ is a probability distribution $\mu(chocolate) = 0.2$, $\mu(vanilla) = 0.5$ and $\mu(stawberry) = 0.3$ we have a lottery $L = [0.2(chocolate),0.5(vanilla),0.3(strawberry)]$

Notice that the concept of a lottery generalises the concept of an outcome. If w is the consequence of a choice s, this corresponds to the lottery where $\mu(w) = 1$, for all $w' \in \Omega$ such that $w \neq w', \mu(w') = 0$. This is what is called a **degenerate lottery**. Degenerate lottery are essentially representations of "certain outcomes".

Similar to the previous case, we can define a preference relation $\succeq$ over a set of lotteries L. Again, we can define a utility function as a function that represents the preferences of an agent over their set of lotteries. **Basically, we will make the agents express their preference over the different possible lotteries.**

We also define the notion of a **compound lottery**, (ie a lottery of lotteries). Given a set $L = \{L_1, \ldots, L_n\}$ of lotteries, a compound lottery is a probability distribution μ over L and is given by $L^* = [\mu(L_1), \ldots, \mu(L_n)]$

Imagine the path you choose to go to work changes depending on whether it rains or not. When you choose a path it takes different times to reach your destination depending on the circumstances. Let Ω be the possible minutes it takes to go to work. Define two lotteries L_{rain} and L_{sun} over Ω that represent the probability of reaching work in a certain amount of time. We define a **compound lottery** $L^* = [pL_{rain}, (1-p)L_{sun}]$ where p is the probability it will rain. L^* is a compound lottery that described the uncertainty of reaching the destination in a certain time taking into account the weather.

Expected Utility:

- given a set of outcomes Ω and a utility function $u : \Omega \to R$, the expected utility of a lottery L, with probability distribution μ is given by: $EU(L) = \sum_{w \in \Omega} u(w)\mu(w)$. EU is the expected value of the function u given the probability distribution μ. EU **is the average utility we could expect from the lottery.**

We have introduced preferences over lotteries, utility functions over outcomes and expected utility. But how are these connected? **What is the relation between preferences over lotteries and expected utility? Notice these are two very different concepts. A preference over a lottery is a qualitative concept (not a numerical one - here the order/preference is important). Expected utilities on the other hand are numerical representations. To connect the two we need to introduce certain axioms about the properties of preference relations. These axioms are beyond the scope of the module.** Von Neumann and Morgenstern discovered a link between preferences over lotteries and expected utility based on utility functions of outcomes. Their result connects the ordinal representation of preferences with the numerical concept of an expected utility. Preferences are required to satisfy the following axioms:

1. Axiom **Continuity**: for every lotteries $L_1 \succ L_2 \succ L_3$, there exists $\alpha, \beta \in [0,1]$ such that $[\alpha L_1, (1-\alpha)L_3] \succ L_2 \succ [\beta L_1, (1-\beta)L_3]$
2. Axiom **Independence** - for all lotteries L_1, L_2, L_3 and $\alpha \in [0,1]$: $L_1 \succeq L_2 \; IFF [\alpha L_1, (1-\alpha)L_3] \succeq [\alpha L_2, (1-\alpha)L_3]$

Theorem:
Let $\hat{L}$ be the set of compound lotteries over a finite set Ω and let $\succeq$ be a preference relation over $\hat{L}$. Then, the following are equivalent:

- ($\succeq$ satisfied **continuity** and **independence**) *IFF* (There exists a utility function over Ω such that for all $L_1, L_2 \in \hat{L}$: $L_1 \succeq L_2 \; IFF \; EU(L_1) \succeq EU(L_2)$ so Expected Utility gives us a numerical representation of qualitative concept as preference over lotteries)

In other words, if $\succeq$ (preference relation over compound lotteries) satisfies the two axioms, then we can use Expected Utility to numerically represent preference over lotteries.

3.2 Strategic-Form Games

We are going to talk about strategic-form noncooperative games. This is the best-known class of games. In non-cooperative games, players act alone, do not make joint decision and pursue their own goals. **We always assume our games have finitely many players $P_1, \ldots, P_n$**. In strategic-form games, each player P_i has different **choices called strategies**. Players choose simultaneously one among their strategies and their combined choices determine different outcomes. Each player will have their own preference over these outcomes which will be represented by a utility function.

Example: Prisoner's dilemma
Two criminals (Prisoner 1 and prisoner 2) are arrested. Each prisoner is in solitary confinement with no means of communicating with the other. The prosecutors lack sufficient evidence to convict the pair on the principal charge, but they have enough to convict both on a lesser charge. Simultaneously, the prosecutors offer each prisoner a bargain. Each prisoner is given the opportunity either to betray the other by testifying that the other committed the crime or to cooperate with the other by remaining silent. Each prisoner has 2 choice: betray their fellow criminal and confess (B) or cooperate with the other prisoner (C). If they both choose to betray each other, they serve 3 years in prison. If they cooperate and stay silent, both of them will only serve one year in prison. If one betrays the other, and the other stays silent, the traitor will be set free, and the other will serve 6 years.

EXAMPLE: PRISONER'S DILEMMA

Prisoner 2

Prisoner 1	C	B
C	$1y$, $1y$	$6y$, $0y$
B	$0y$, $6y$	$3y$, $3y$

- We can represent the Prisoner's Dilemma in matrix form.
- There are 4 possible outcomes, i.e.

$$\Omega = \{0y, 1y, 3y, 6y\}$$

- We can define a preference relation $\succ_i$ for each player P_i such that

$$0y \succeq_i 1y \succeq_i 3y \succeq_i 6y.$$

Each $\succ_i$ can be represented in terms of a utility function u_i

For instance: $u_i(0y) = 10$, $u_i(1y) = 8$, $u_i(3y) = 5$, $u_i(6y) = 0$

We can represent the game in matrix form with the utility functions representing players' preferences.

More formally, a strategic-form game is a tuple $(N, S_1, \ldots, S_n, u_1, \ldots, u_n)$ where:
- $N = \{1,...,n\}$ is a finite set of players
- S_i is a finite set of strategies for each player i
- $u_1 : S_1 \times \ldots \times S_n \rightarrow R$ is a utility function for player I (so to each possible strategy combination, we assign a real number)

Each $(s_1, \ldots, s_n) \in S_1 \times \ldots \times S_n$ is called a **strategy profile or strategy combination**
Strategy profiles are also denoted by (s_i, s_{-i}) to highlight the strategy of player i. s_{-i} denotes the strategy combination without player i. $s_{-i} = (s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n)$

S_{-i} is the set of all strategy combinations of the form s_{-i} (it excluding strategies for player i)

(In game theory) We assume that players are rational decision-makers and **have complete and common knowledge about each other strategies**, **utilities** and their **rationality**. We are not interested in how players **play** a game (ie descriptive, empirical interpretations). We are not interested in how players **should** play a game (ie define rules/normative interpretation). **We are interested in trying to predict what will happen under the above assumptions** (ie theoretical interpretation). So, if players act rationally, **what outcome will they choose**? We answer the question by defining solution concepts, ie criteria that will allow us to predict the solution of a game under the assumptions we make about players' behaviour.

Dominated Strategies

- the outcome for a player will always depend on the choice of others, but there are situations where one player can make independent choices that will always yield better outcomes.
- For instance, prisoner 1 will always get a higher utility choosing B(betrayal) over C (Cooperation), no matter what Prisoner 2 chooses. In this case, we say that for Prisoner 1, C is strictly dominated by B.

		Prisoner 2	
		C	B
Prisoner 1	C	8, 8	0, 10
	B	10, 0	5, 5

Definition:

- a strategy s_i of player i is strictly dominated if there exists another strategy s_i' of player i that for each strategy vector $s_{-i} \in S_{-i}$ of the other players, $u_i(s_i, s_{-i}) < u_i(s_i', s_{-i})$
- In this case we say that s_i is strictly dominated by s_i'
- **Since we have assumed that all players are rational and also know about each other's rationality => we can then assume that rational players will never play strictly dominated strategies, which can then be eliminated from the game**

For both prisoners, B strictly dominates C. So under our assumption, we can conclude that the outcome of the game will be the strategy combination (B, B). This process is called **iterated elimination of strictly dominated strategies. Whenever we can eliminate strictly dominated strategies, the result is always independent of the order of elimination!**

		Prisoner 2	
		C	B
Prisoner 1	C	8, 8	0, 10
	B	10, 0	5, 5

The problem is that not all games have strictly dominated strategies, and so, we cannot always reach an outcome by elimination. For example, this game cannot always reach an outcome by elimination:

		Player 2	
		C	*D*
Player 1	*A*	1, 2	2, 3
	B	2, 2	2, 0

However, there are strategies that are at least as good as others for some players. If player 1 selects B, the outcome will be as good as selecting A, if not better.

B does not strictly dominate A
Here B does weakly dominate A

A strategy s_i of player i is **weakly dominated** if there exists another strategy s_i' of player i satisfying the following two conditions:

- 1) for every strategy vector $s_{-i} \in S_{-i}$ of the other players, $u_i(s_i, s_{-i}) \leq u_i(s_i', s_{-i})$
- 2) There exists a strategy vector $t_{-i} \in S_{-i}$ of the other players, such that $u_i(s_i, t_{-i}) < u_i(s_i', t_{-i})$
- In this case, we say that s_i is **weakly dominated** by s_i'
- **We can then assume that rational players will never play weakly dominated strategies**, which can then be eliminated from the game.

Similar to strictly dominated strategies, elimination of weekly dominated strategies cannot always be performed. **In addition, different from strictly dominated strategies, the order of elimination does matter and we can get different results (see example of 2 different iterations on next page)**

		Player 2		
		D	*E*	*F*
Player 1	*A*	1, 2	2, 3	0, 3
	B	2, 2	2, 1	3, 2
	C	2, 1	0, 0	1, 0

Player 2

	D	E	F
Player 1 A	1, 2	2, 3	0, 3
Player 1 B	2, 2	2, 1	3, 2
Player 1 C	2, 1	0, 0	1, 0

Player 2

	D	E	F
Player 1 A	1, 2	2, 3	0, 3
Player 1 B	2, 2	2, 1	3, 2
Player 1 C	2, 1	0, 0	1, 0

Player 2

	D	E	F
Player 1 B	2, 2	2, 1	3, 2
Player 1 C	2, 1	0, 0	1, 0

Player 2

	D	E	F
Player 1 B	2, 2	2, 1	3, 2
Player 1 C	2, 1	0, 0	1, 0

Player 2

	D	E	
Player 1 B	2, 2	2, 1	
Player 1 C	2, 1	0, 0	

Player 2

	D	E	
Player 1 B	2, 2	2, 1	
Player 1 C	2, 1	0, 0	

Player 2

	D	E	
Player 1 B	2, 2	2, 1	

Player 2

	D	E	
Player 1 B	2, 2	2, 1	

Player 2

	D		
Player 1 B	2, 2		

Player 2

	D	E	F
Player 1 A	1, 2	2, 3	0, 3
Player 1 B	2, 2	2, 1	3, 2
Player 1 C	2, 1	0, 0	1, 0

Player 2

	D	E	F
Player 1 A	1, 2	2, 3	0, 3
Player 1 B	2, 2	2, 1	3, 2
Player 1 C	2, 1	0, 0	1, 0

Player 2

	D	E	F
Player 1 A	1, 2	2, 3	0, 3
Player 1 B	2, 2	2, 1	3, 2

Player 2

	D	E	F
Player 1 A	1, 2	2, 3	0, 3
Player 1 B	2, 2	2, 1	3, 2

Player 2

		E	F
Player 1 A		2, 3	0, 3
Player 1 B		2, 1	3, 2

Player 2

		E	F
Player 1 A		2, 3	0, 3
Player 1 B		2, 1	3, 2

Player 2

			F
Player 1 A			0, 3
Player 1 B			3, 2

Player 2

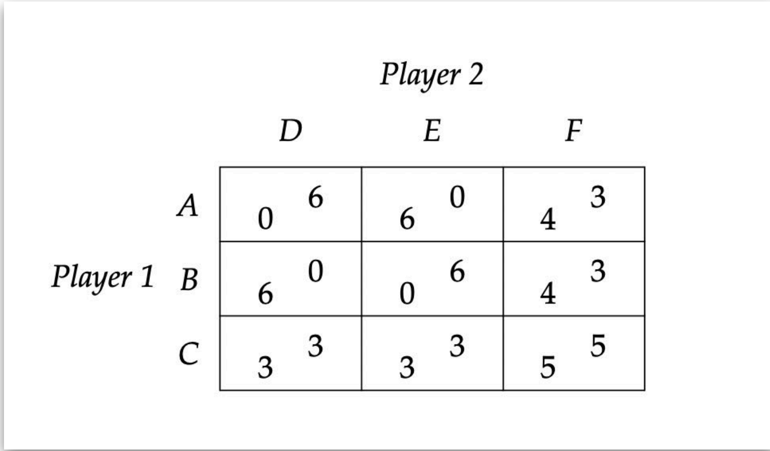
			F
Player 1 A			0, 3
Player 1 B			3, 2

Player 2

			F
Player 1 B			3, 2

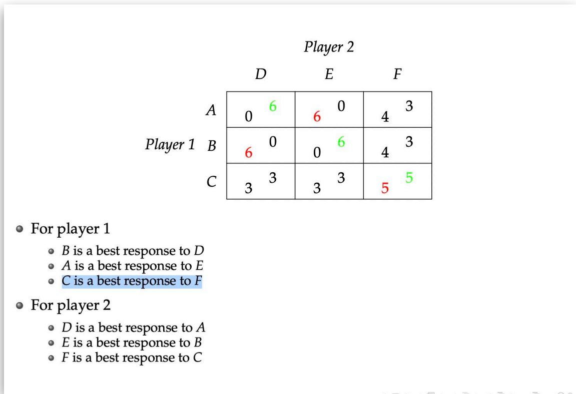
3.3 Pure Nash Equilibria

There are games where we cannot perform elimination of dominated strategies. We need different solution concepts. The most important one is the concept of stability: the Nash equilibrium.



To understand the concept of a Nash equilibrium we need the **concept of a best response**. A player's best response to a strategy profile is a choice that gives the player the highest utility. The best response does not have to be unique. More formally:

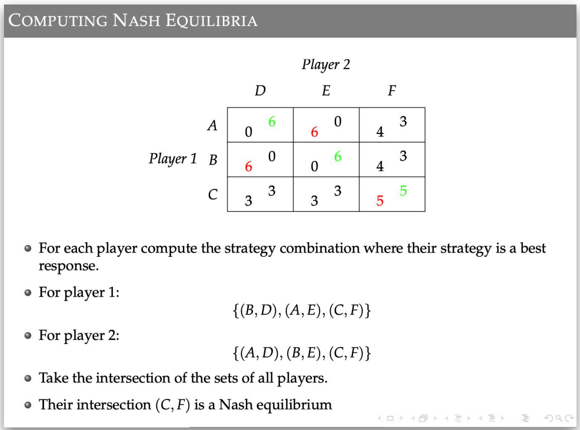
Let s_{-i} be a strategy vector for all the players except i. Player i's strategy s_i, is called a best response to s_{-i} if $u_i(s_i, s_{-i}) = max_{s_i' \in S_i} u_i(s_i', s_{-i})$

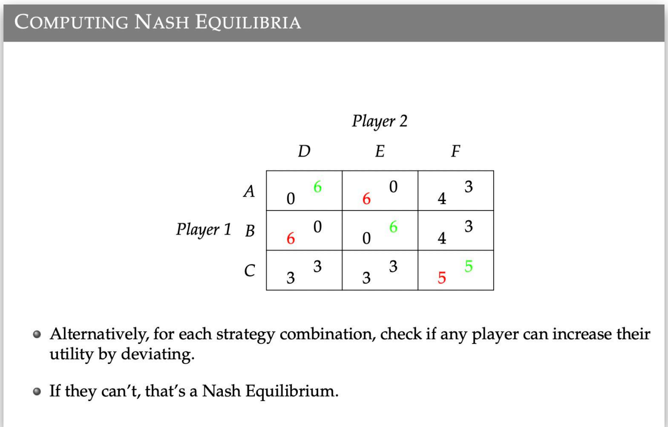


The strategy combination (C, F) is such that the strategies are best response to each other. If players selects this combination, none of them will benefit from changing their choice, because they have chosen a best response. This is a situation of stability, in fact (C,F) is an example of a Nash equilibrium.

A strategy combination $(s_1, \ldots, s_n)$ is a Nash equilibrium if s_i is a best response to s_{-i} for every player $i \in N$.

Determining the existence of a Nash equilibrium for a strategic form game is in logarithmic space





Nash equilibria might not be unique. Coordination games are examples of games with multiple Nash equilibria. Equilibria arise when players coordinate on the same strategy.

Player 1 \ *Player 2*	*C*	*D*
A	10, 10	0, 8
B	8, 0	7, 7

Player 1 \ *Player 2*	*H*	*T*
H	1, −1	−1, 1
T	−1, 1	1, −1

Not all games have Nash equilibria. **Matching Pennies** is one example of this. Two players toss a penny simultaneously. If the outcomes match, player 1 keeps both pennies. If the outcomes don't match, it is player 2 who gets to keep both coins. To be in a situation where equilibria always exist we will need the concept of a mixed strategy (more on this later).

Iterated Elimination and Nash Equilibria

Let $(s_1, \ldots s_n)$ be a strategy profile obtained from iterated elimination of **strictly dominated** strategies. **Then** $(s_1, \ldots s_n)$ **is a Nash equilibrium**. Moreover, $(s_1, \ldots s_n)$ is the unique equilibrium of the game. **Iterated elimination of strictly dominated strategies does not eliminate equilibria from the game which is really nice.**

Prisoner 1 \ Prisoner 2	*C*	*B*
C	5, 5	0, 10
B	10, 0	8, 8

P1 \ P2	*L*	*R*
T	0, 0	2, 1
B	3, 2	1, 2

- Equilibria: (T, R) and (B, L).
- L is weakly dominated by R and, after eliminating L, B is weakly dominated by T.
- The result of eliminating weakly dominated strategies is (T, R) and equilibrium (B, L) is lost.

Given a game G, let G* be he game obtained by iterated elimination of **weakly dominated strategies**. The set of equilibria of G* is a subset of the set of equilibria of G. This means that iterated elimination of weakly dominated strategies can result in the elimination of some (if not all) the equilibria of the original game.