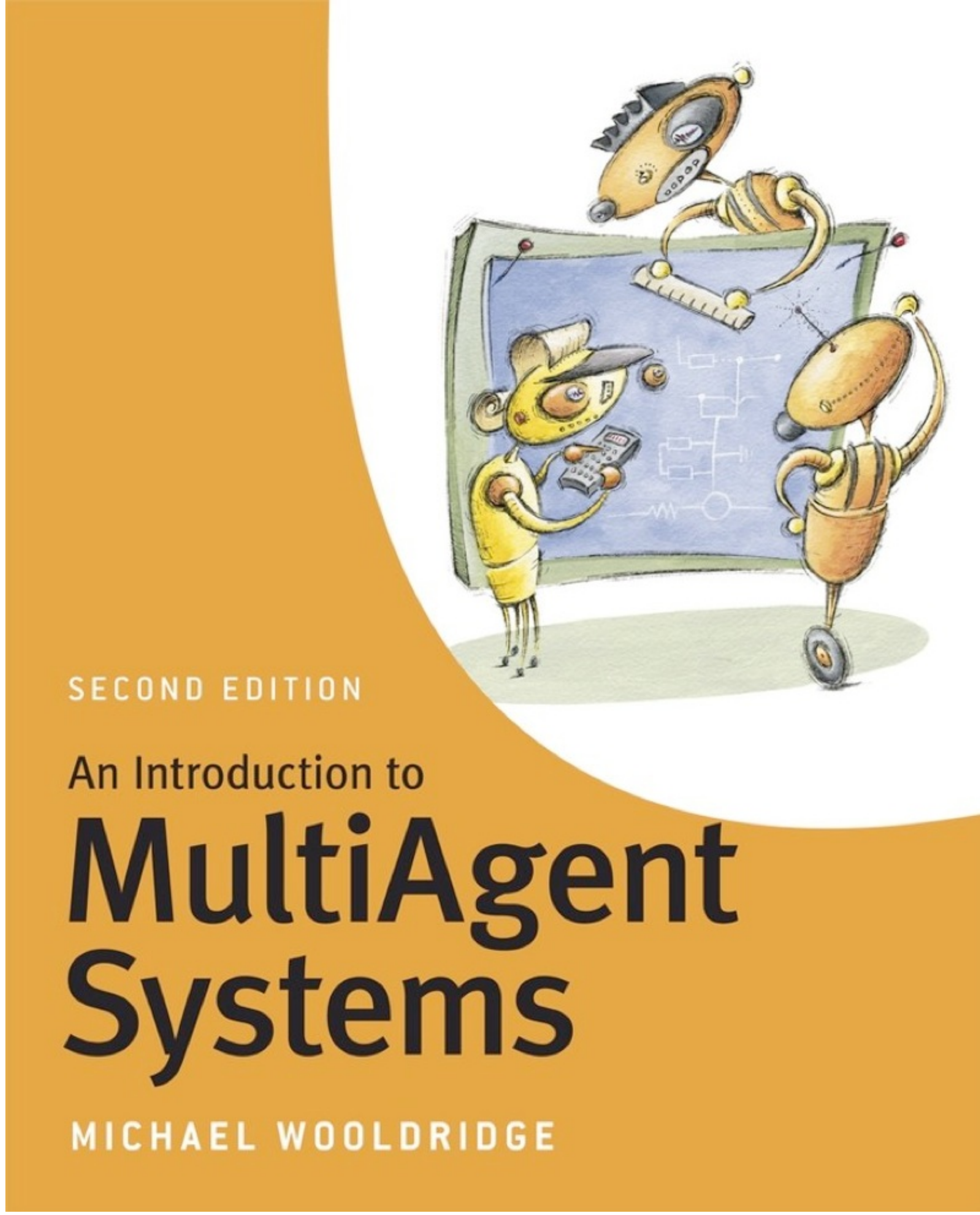
COMP310
Multi-Agent Systems
Chapter 15 - Bargaining / Negotiation

Dr Terry R. Payne
Department of Computer Science

UNIVERSITY OF LIVERPOOL



Overview

- How do agents ***reach agreements*** when they are self interested?
  - In an extreme case (zero sum encounter) no agreement is possible — but in most scenarios, there is potential for ***mutually beneficial agreement on matters of common interest***.
- The capabilities of: ***negotiation*** and ***argumentation*** are central to the ability of an agent to reach such agreements.
- This chapter will talk about negotiation and in the next chapter we'll go on to cover argumentation.

Mechanisms, Protocols, and Strategies

- Negotiation is governed by a particular ***mechanism***, or ***protocol***.
  - The mechanism defines the "rules of encounter" between agents.
- ***Mechanism design*** is designing mechanisms so that they have certain desirable properties.
  - Properties like Pareto efficiency
- Given a particular protocol, how can a particular ***strategy*** be designed that individual agents can use?

Auction vs Negotiation

- Auctions are ***only*** concerned with the allocation of goods: richer techniques for reaching agreements are required.

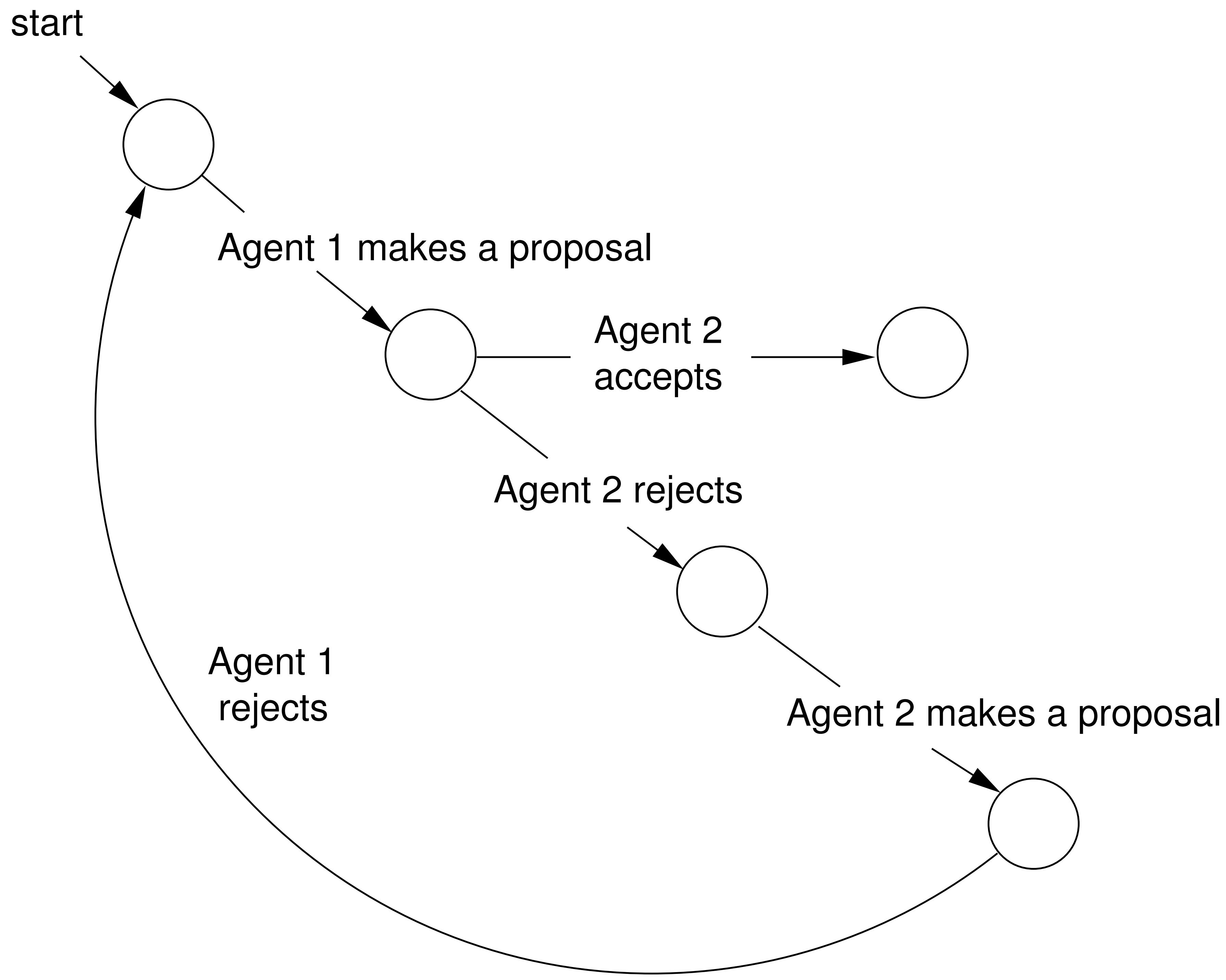
- ***Negotiation*** is the process of reaching agreements on matters of common interest.
  - Any negotiation setting will have four components:
    - A ***negotiation set***: possible proposals that agents can make.
    - A ***protocol***: which defines what legal proposals an agent can make, based on prior negotiations.
    - A collection of ***strategies***, one for each agent, which are private.
    - A ***rule*** that determines ***when a deal has been struck*** and ***what the agreement deal*** is.
  - Negotiation usually proceeds in a series of rounds, with every agent making a proposal at every round.

Auction vs Negotiation

- There are a number of aspects of negotiation that make it complex.
  - Multiple issues:
    - Number of possible deals is exponential in the number of issues.
      - (Like the number of bundles in a combinatorial auction)
    - Hard to compare offers across multiple issues
      - The car salesman problem
  - Multiple agents
    - One-to-one / Many-to-one / Many-to-many negotiation
- At the simple end there isn't much to distinguish negotiation from auctions.

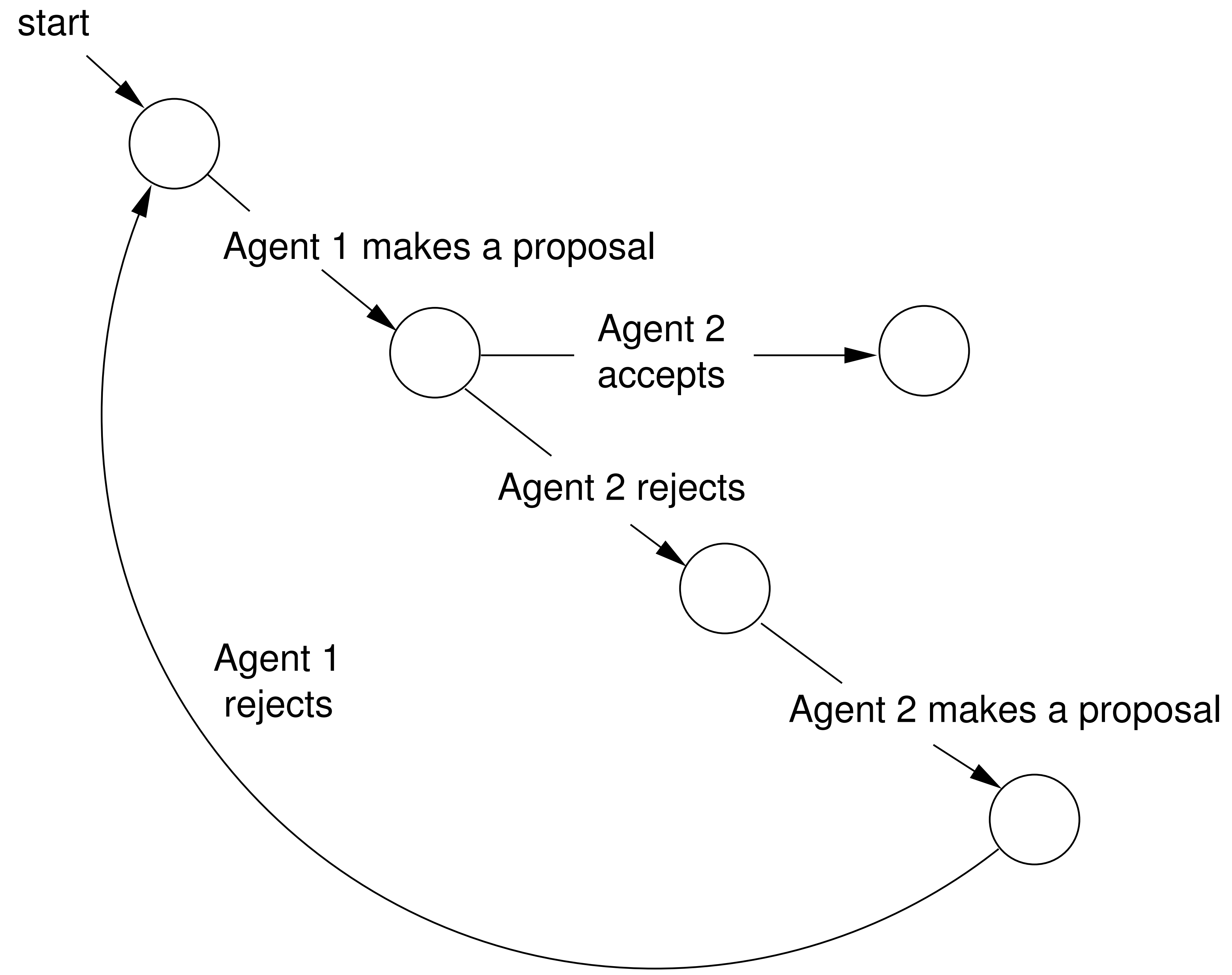
Rubinstein’s alternating offers

- We will start by looking at Rubinstein’s ***alternating offers*** model.
  - This is a one-to-one protocol.
  - Agents are 1 and 2, and they negotiate over a series of rounds:
    - In ***round 0***, Agent 1 makes an offer x^0.
      - Agent 2 either accepts A, or rejects R.
  - If the offer is accepted, then the deal is implemented.
    - If not, we have ***round 1***, and Agent 2 makes an offer, etc.



Rubinstein's alternating offers

- The rules of the protocol will not guarantee that agreement will always be reached.
  - Agents could just keep rejecting offers.
- If there is ***no agreement***, we say the result is the ***conflict deal*** Θ.
- We make the following basic assumptions:
  - ***Disagreement is the worst outcome***
    - Both agents prefer any agreement to none.
  - ***Agents seek to maximise utility***
    - Agents prefer to get larger utility values
- With this basic model, we get some odd results.



Imagine we are dividing a pie...

- Model this as some resource with value 1, that is divided into two parts.
  - Each part is between 0 and 1.
    - The two parts sum to 1 so a proposal is $(x, 1 - x)$
- The set of possible deals is:
  - $\{(x, 1 - x) : 0 \leq x \leq 1\}$
- If you are Agent 1, what do you offer?

If we had …

- What if we had 1 round?
  - Let's assume that we will only have one round.
  - *Ultimatum game*
    - Agent 1 has all the power.
    - If Agent 1 proposes (1, 0), then this is still better for Agent 2 than the conflict deal.
  - Agent 1 can do no better than this either.
  - So we have a Nash equilibrium.

- What if we had 2 rounds?
  - The power passes to Agent 2.
    - Whatever Agent 1 proposes, Agent 2 rejects it.
    - Then Agent 2 proposes (0, 1).
  - Just as before…
    - …this is still better for Agent 1 than the conflict deal and so it is accepted.
  - This will happen any time there is a fixed number of rounds.

What about an indefinite number of rounds?

- What if we have an indefinite number of rounds?
  - Let's say that Agent 1 uses this strategy:
    - *Always propose* (1, 0) *and always reject any offer from Agent 2*
  - How should Agent 2 respond?
- If Agent 2 rejects, then there will never be agreement.
    - We end up with the conflict deal
  - So Agent 2 should accept.
    - And there is no point in not accepting on the first round.

In fact, whatever $(x, 1 - x)$ offer that agent 1 proposes here, immediate acceptance is the Nash equilibrium as long as Agent 2 **knows** what Agent 1's strategy is.

- There are thus an infinite number of Nash Equilibria.
- All are Pareto optimal.

Impatient Players

- Since we have an infinite number of Nash equilibria
  - the ***solution concept of NE is too weak*** to help us.
- Can get unique results if we take time into account.
  - For any outcome x and times $t_2 > t_1$, both agents prefer x at time t_1.
- A standard way to model this impatience is to discount the value of the outcome.

Impatient Players

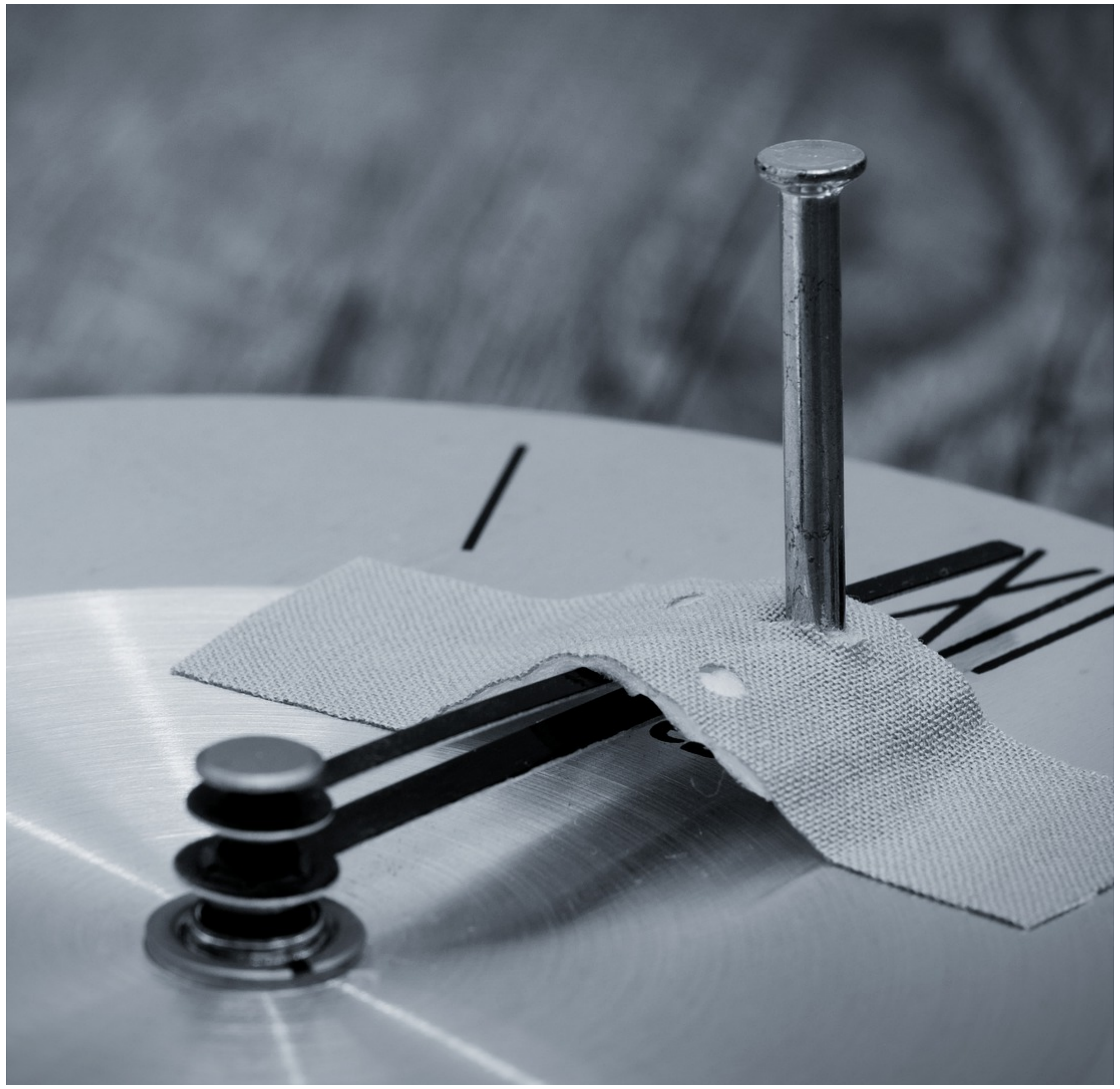
Each agent has a **discount factor** δ_i where $i \in \{1, 2\}$, and $0 \leq \delta < 1$. The closer δ_i is to 1, the more patient the agent is.

If agent i is offered x, then the value of the slice is:

- x at time 0.
- $\delta_i x$ at time 1.
- $(\delta_i * \delta_i)x = \delta_i^2 x$ at time 2.
- $(\delta_i * \delta_i * \delta_i)\mathrm{x} = \delta_i^3 x$ at time 3.
- …
- $\delta^k x$ at time k

Now we can make some progress with the fixed number of rounds.

Impatient Players



- A ***1-round*** game is an ***ultimatum*** game.
- A ***2-round*** game means Agent 2 can play as before, but if so, will only get δ_2.
  - Gets the whole pie, but it is worth less.
- Agent 1 can take this into account.
  - If Agent 1 offers $(1-\delta_2, \delta_2)$ then Agent 2 might as well accept — can do no better.
  - This is now a Nash equilibrium.

Impatient Players

- In the general case, Agent 1 makes the proposal that gives Agent 2 what Agent 2 would be able to enforce in the second round.
  - Agent 1 gets: $\dfrac{1-\delta_2}{1-\delta_1\delta_2}$
  - Agent 2 gets: $\dfrac{\delta_2(1-\delta_1)}{1-\delta_1\delta_2}$

Division of the Pie with Discount Fctors

Discount factor δ_1

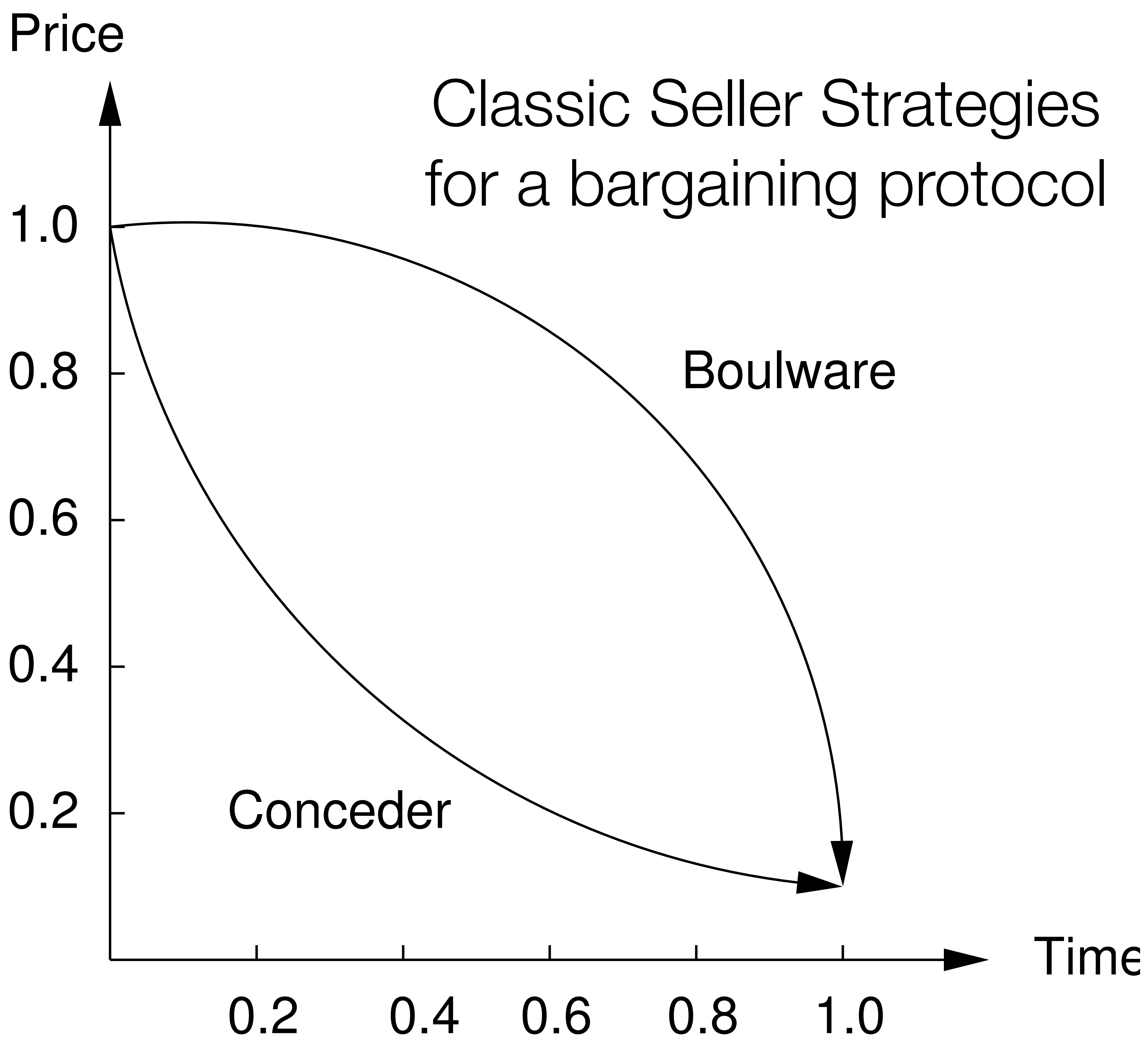
	0.8	**0.6**	**0.4**	**0.2**
0.8	(0.556,0.444)	(0.769,0.231)	(0.882,0.118)	(0.952,0.048)
0.6	(0.385,0.615)	(0.625,0.375)	(0.789,0.211)	(0.909,0.091)
0.4	(0.294,0.706)	(0.526,0.474)	(0.714,0.286)	(0.870,0.130)
0.2	(0.238,0.762)	(0.455,0.545)	(0.652,0.348)	(0.833,0.167)

Discount factor δ_2

- Note that the more patient either agent is (i.e. as δ increases), the more pie they get.

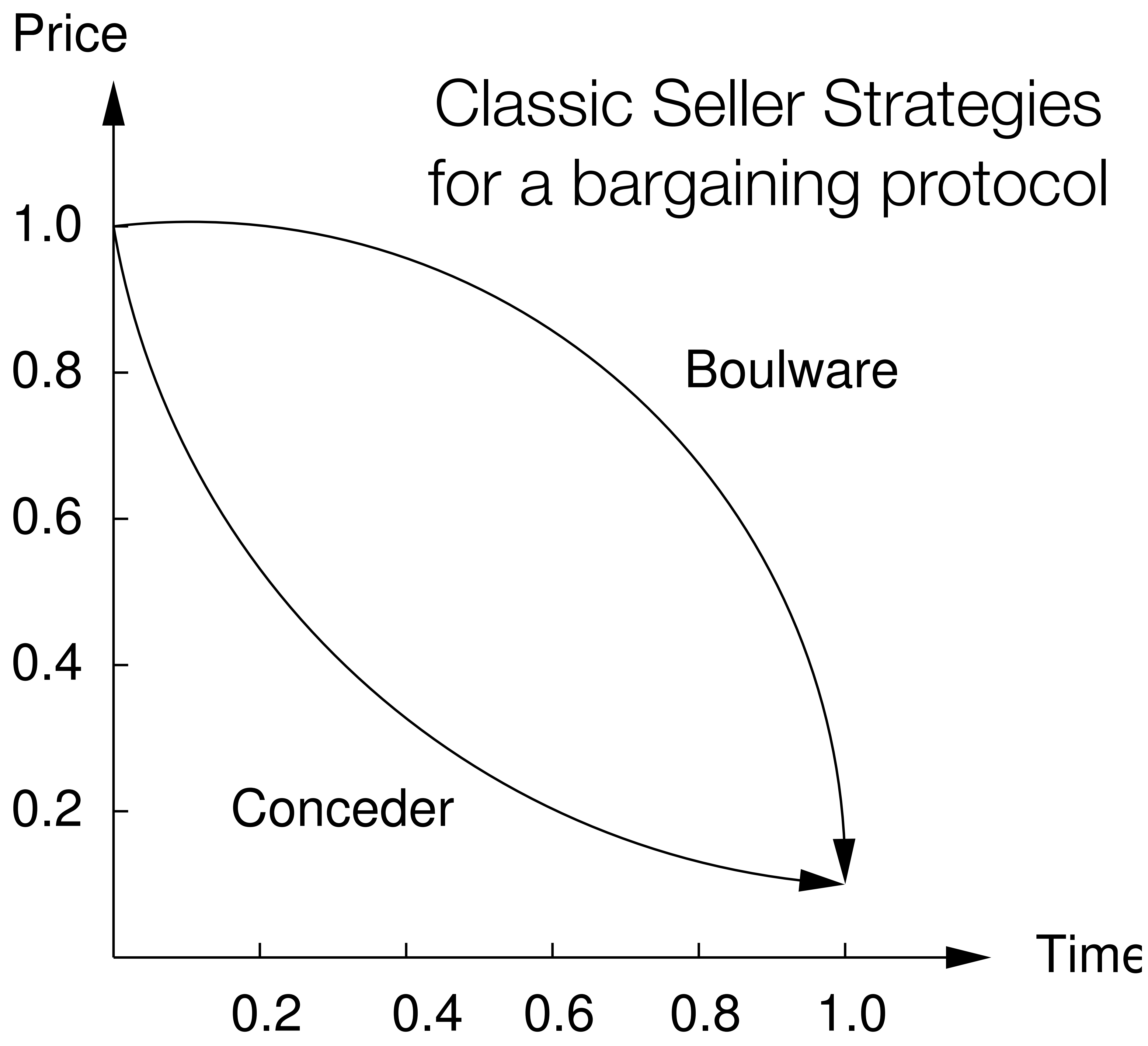
Negotiation Decision Functions

- The approach we just talked about relies on ***strategic*** thinking about the other player.
  - How can each player maximise its own share of the pie
- Instead of being strategic, a simpler approach is to use a negotiation decision function
  - This is time dependent, and shapes the approach to bargaining
- This simplifies the need for determining how to negotiate
  - But fails to take into account the opponent’s actions or behaviours



Negotiation Decision Functions

- Boulware
  - Very ***slow initial decrease*** in price until close to deadline and then an exponential decrease.
- Conceder
  - ***Makes its concessions early***, and makes fewer concessions later in the negotiation.



Negotiation in Task-Oriented Domains

"... Imagine that you have three children, each of whom needs to be delivered to a different school each morning. Your neighbour has four children, and also needs to take them to school.

Delivery of each child can be modelled as an indivisible task. You and your neighbour can discuss the situation, and come to an agreement that it is better for both of you (for example, by carrying the other's child to a shared destination, saving him the trip).

There is no concern about being able to achieve your task by yourself. The worst that can happen is that you and your neighbour won't come to an agreement about setting up a car pool, in which case you are no worse off than if you were alone. You can only benefit (or do no worse) from your neighbour's tasks.

Assume, though, that one of my children and one of my neighbours's children both go to the same school (that is, the cost of carrying out these two deliveries, or two tasks, is the same as the cost of carrying out one of them). It obviously makes sense for both children to be taken together, and only my neighbour or I will need to make the trip to carry out both tasks ..."

Task-Oriented Domains (TODs) Defined

- A task-oriented domain (TOD) is a triple $\langle T, Ag, c\rangle$
  - T is the (finite) set of all possible ***tasks***;
  - $Ag = \{1,...,n\}$ is set of ***participant agents***;
  - The ***cost*** of executing each subset of tasks is defined as: $c : 2^T \rightarrow \mathbb{R}^+$
    - The cost function should be ***monotonic***: i.e. if $T_1 \subseteq T_2$ are subsets of tasks, then $c(T_1) \leq c(T_2)$
    - The ***cost of doing nothing is zero***: i.e. $c(\emptyset) = 0$
  - An ***encounter*** is a collection of tasks $\langle T_1 ,..., T_n\rangle$
    - where $T_i \subseteq T$ for each $i \in Ag$.
    - This is where agents can reach a deal by reallocating the tasks amongst themselves

Deals in TODs

- Assuming that we have only two agents (1 and 2), and given the encounter $\langle T_1, T_2\rangle$, a deal will be an allocation of the tasks $T_1 \cup T_2$.
  - The ***cost*** to agent i of deal $\delta = \langle D_1, D_2\rangle$ is $c(D_i)$
    - This is denoted $cost_i(\delta)$.
  - The ***utility*** of deal δ to agent i is $utility_i(\delta) = c(T_i) - cost_i(\delta)$
    - The utility represents ***how much the agent has to gain*** from the deal
    - If the ***utility is negative then the agent is worse off*** from the deal
    - The deal δ is ***individually rational*** if it gives positive utility.
  - What if the agents fail to reach a deal?
    - The ***conflict deal***, Θ, is the deal $\langle T_1, T_2\rangle$ consisting of the tasks originally allocated;
      - i.e. $utility_i(\Theta) = 0$ for all $i \in Ag$

Dominance in Deals

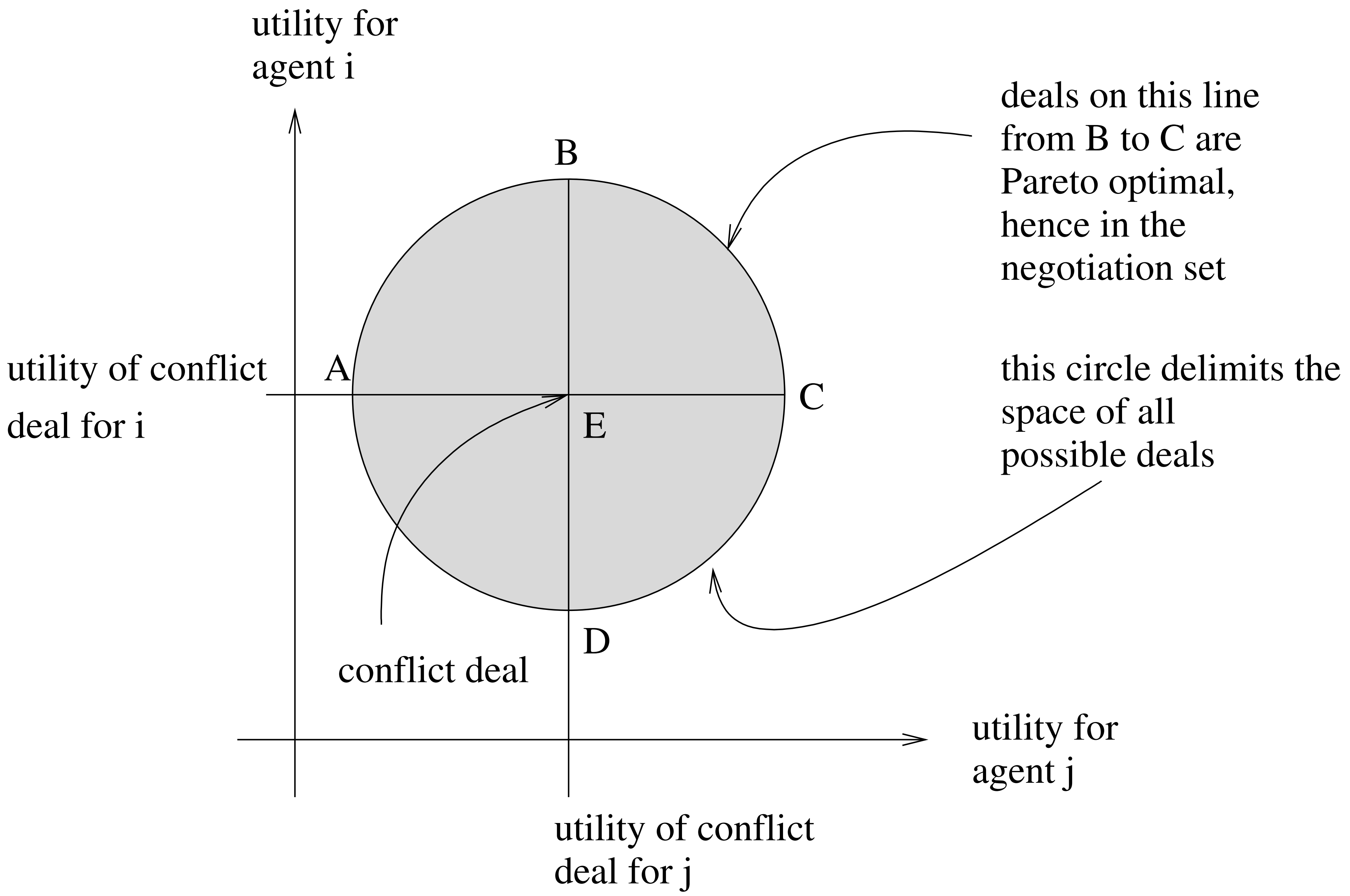
- A deal δ_1 is said to ***dominate*** δ_2 (i.e. $\delta_1 > \delta_1$) if the following hold:
  - Deal δ_1 is at least as good for every agent as δ_2:
    - $\forall i \in \{1,2\}, utility_i(\delta_1) \geq utility_i(\delta_2)$
  - Deal δ_1 is better for some agent than δ_2:
    - $\exists i \in \{1,2\}, utility_i(\delta_1) > utility_i(\delta_2)$
- Thus, if one deal (δ_1) dominates another (δ_2), then all *"reasonable"* participants would prefer that first deal (δ_1)
  - A deal ***that is not dominated is Pareto optimal***.
- A deal (δ_1) is ***"individual rational"*** if it weakly dominates the conflict deal
  - I.e. if it at least as good for every agent as the conflict deal (see first condition above)

The Negotiation Set

- The set of deals over which agents negotiate are those that are said to be:
  - Individual rational
  - Pareto optimal.

- ***Individual rational***: agents won't be interested in deals that give negative utility since they will prefer the conflict deal.

- ***Pareto optimal***: agents can always transform a non-Pareto optimal deal into a Pareto optimal deal by making one agent happier and none of the others worse off.



The Monotonic Concession Protocol

- Rules of this protocol are as follows. . .
  - Negotiation proceeds in rounds u where $u \geq 0$.
  - On round u=1:
    - Agents simultaneously propose a deal from the negotiation set.
    - Agreement is reached if one agent finds that the deal proposed by the other is at least as good or better than its proposal.
    - If no agreement is reached, then negotiation proceeds to another round of simultaneous proposals.
  - In round $u + 1$:
    - No agent is allowed to make a proposal that is less preferred by the other agent than the deal it proposed at time u.
  - If neither agent makes a concession in some round $u > 0$, then negotiation terminates, with the conflict deal.

The Monotonic Concession Protocol

- With this protocol, negotiation is guaranteed to end after a finite number of rounds
  - Either the agents will reach an agreement, or a round occurs where neither agent concedes.
  - No guarantees however that an agreement will finish quickly
  - Number of possible deals is $O(2^{|T|})$
    - I.e the number of rounds is exponential in the number of tasks allocated

The Zeuthen Strategy

- Three problems:
  - What should an ***agent's first proposal*** be?
    - Its ***most preferred*** deal
  - On any given round, ***who should concede***?
    - The agent ***least willing to risk conflict***.
      - An agent will be ***less willing to risk conflict*** if the difference in utility between its current proposal and the conflict deal is ***high***.
  - If an agent concedes, then ***how much should it concede***?
    - Just enough to change the balance of risk.

Willingness to Risk Conflict

- Suppose you have conceded a ***lot***. Then:
  - Your proposal is now near to conflict deal.
  - In case conflict occurs, you are not much worse off.
  - You are ***more willing to risk conflict***.

- An agent will be ***more willing to risk conflict*** if the difference in utility between its current proposal and the conflict deal is ***low***.

Nash Equilibrium Again...

- The Zeuthen strategy is in Nash equilibrium: under the assumption that one agent is using the strategy the other can do no better than use it himself. . .

This is of particular interest to the designer of automated agents. It does away with any need for secrecy on the part of the programmer. An agent's strategy can be publicly known, and no other agent designer can exploit the information by choosing a different strategy. In fact, it is desirable that the strategy be known, to avoid inadvertent conflicts.

Deception in TODs

- Deception can benefit agents in two ways:
  - Phantom and Decoy tasks.
    - Pretending that you have been allocated tasks you have not.
    - This can be prevented by ensuring that tasks are ***verifiable***
  - Hidden tasks.
    - Pretending not to have been allocated tasks that you have.
      - If you have one task that is similar to that of another agent, then it might propose that you take on its similar task
      - This may incur a small additional task
      - If however you don't mention it, then the proposal may not be made.

Summary

- This lecture started to look at different mechanisms for reaching agreement between agents.
    - In particular we looked at negotiation, where agents make concessions and explore tradeoffs.
- We looked at negotiations about the division of resources.
    - Ultimatum game and its variants
- We also looked at negotiation in task-oriented domains where agents can find synergies between tasks and exploit these to reach agreement.
- In the final lectures, we will go on to talk about argumentation, another family of techniques for reaching agreement.

Class Reading (Chapter 15):

"Negotiation and cooperation in Multi-agent Environments", S.Kraus. AI Journal, 94(1-2): 79-98 (1997).

This article provides an overview of negotiation techniques for multi agent systems. It provides a number of pointers into the research literature, and will be particularly useful for mathematically oriented students.