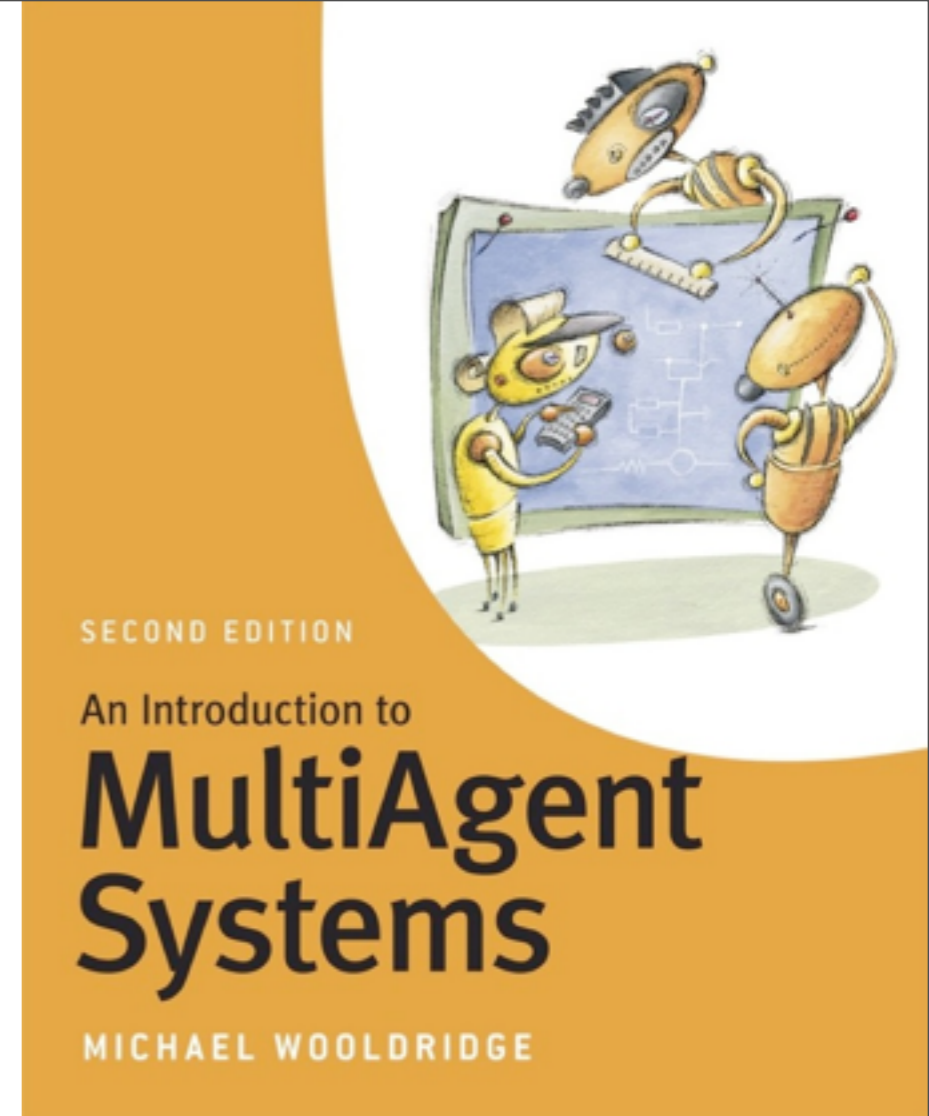


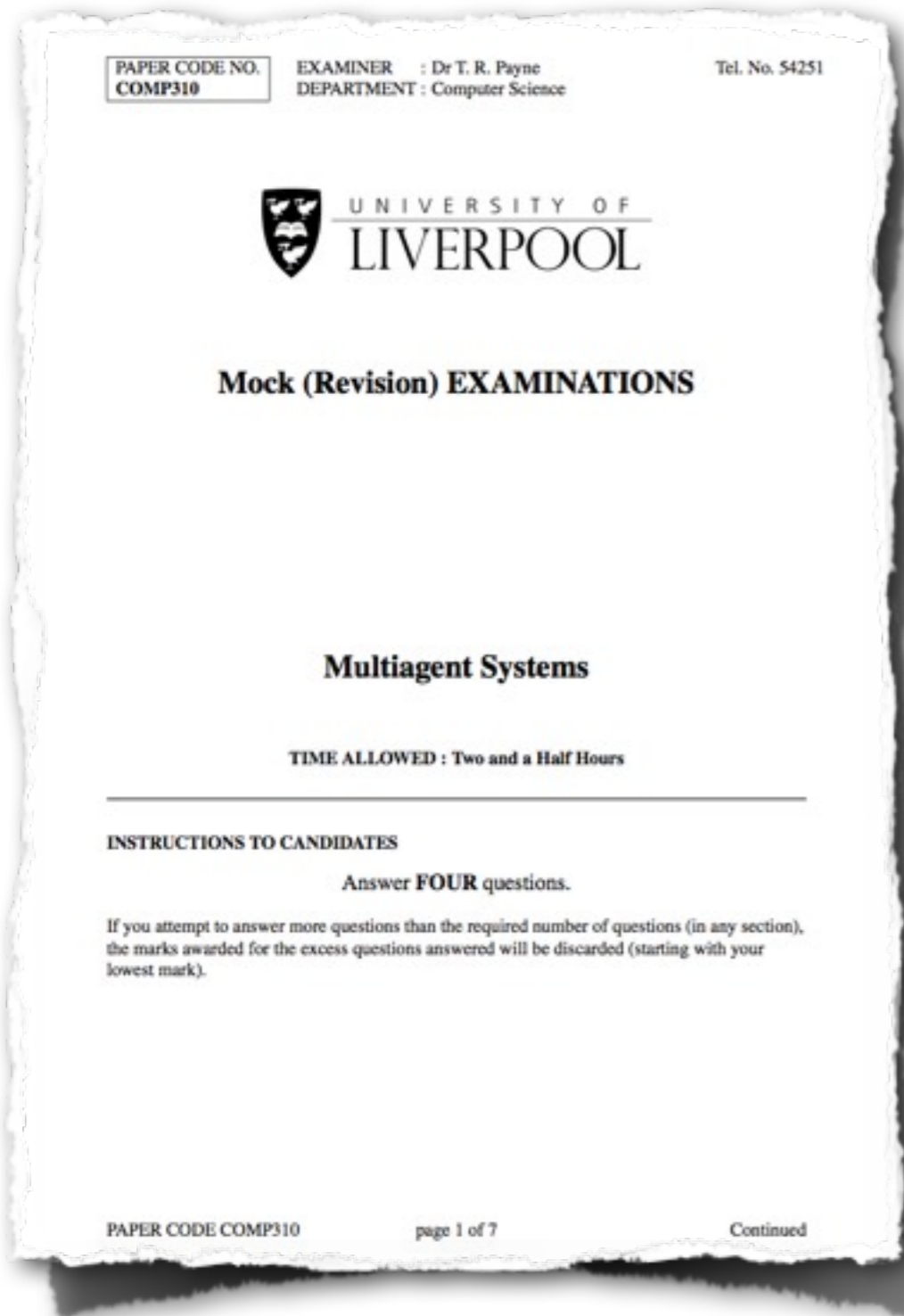
COMP310

MultiAgent Systems

Mock Exam Solutions



Exam Paper and Rubric



- The exam paper format has remained the same for the last few years
- Five Questions
 - Complete four of the five questions, each worth 25 marks
 - If all questions are attempted, the four questions with the highest marks will be considered.
- Exam is 2 1/2 hours long

Coverage

- Any of the material taught could appear in the exam
- Questions may be
 - Book work (what can you tell me about ...)
 - Problem solving (given this, what is that)
- A Mock Paper and Past Exam Papers are available
 - See course website
 - The solutions are provided for the Mock Paper...
 - ...but no solutions will be given out for past exam papers

Question 1.a & 1.b

1. (a) The *deliberative* or *symbolic* paradigm is historically the dominant approach to building autonomous intelligent agents. There are two fundamental problems to be solved if one aims to build such agents. Describe what you understand by these problems.
(4 marks)
- (b) Explain the key commonalities and differences between the multi agent systems paradigm and the object-oriented programming paradigm.
(8 marks)

Q1.a answer

The deliberative or symbolic paradigm relies on the use of a symbolic representation of the world, that can exploit logical formulae and theorem proving to support reasoning about the world. Thus, given some goal and knowledge about both the environment and its capabilities, an agent can determine a set of actions that can be performed for it to reach that goal. However, there are two fundamental problems that need to be resolved if such an agent is to be rebuilt: The transduction problem, and the representation/reasoning problem. The transduction problem relates to the challenge of translating the real world into an accurate and adequate symbolic description that can be used by the agent, and also that this representation can be generated in time for that description to be useful. For example, using a camera to track the ball in a game of robo-soccer is only useful if the representation of the location or trajectory of the ball is still valid when the agent comes to make use of this information. The second problem - the representation / reasoning problem, is concerned with identifying a suitable compact symbolic representation that supports tractable reasoning. As complex logics and/or reasoning algorithms can be computationally complex, it is important that a compact representation is found such that when an agent reasons about its actions in the world, it does this in a timely manor.

Note that this solution is quite detailed - normally I wouldn't expect an answer as long as this for 4 marks. It is a good idea to always start with one or two sentences giving a little background, before then giving your answer, to help give context, and to focus your thoughts.

Q1.b answer

Agents and Objects have a number of things in common. Objects have the notion of having instances which encapsulates some state. Objects communicate via message passing, or can call methods that belong to other objects. And the methods correspond to a set of operations that may be performed on this state. Whilst agents may initially seem to be similar, there are a number of key differences:

- Agents are autonomous! When an object receives a message, or has one of its methods called, it will then simply perform the relevant actions; whereas agents typically embody the notion of utility, which they can use to determine whether or not it is in the agents own interest to perform some action.*
- In addition, agents are "smart"; they can be capable of flexible behaviour, such as being reactive to changes in the environment, or by interacting with other agents, can be social, taking into account how other agents act (a good example of this is in Axelrod's tournament, where agents that reflect their peer's performance in an Iterative Prisoners Dilemma game do better than adopting other strategies, and thus if they all cooperate, they help each other). Finally, they are active - they maintain some level of active control.*

This solution broadly covers the material in the slides, as well as some additional background gained from knowing about object-oriented programming. Note that although the question does not ask about Axelrod's tournament, it is used to support the argument. Often knowledge about other parts of the module could be used to support your explanation, especially if you are not sure that you have explained things as well as you could have...

Question 1.c

(c) Consider the environment $Env_1 = \langle E, e_0, \tau \rangle$ defined as follows:

$$E = \{e_0, e_1, e_2, e_3, e_4, e_5, e_6\}$$

$$\tau(e_0 \xrightarrow{\alpha_0}) = \{e_1, e_2, e_3\}$$

$$\tau(e_0 \xrightarrow{\alpha_1}) = \{e_4, e_5, e_6\}$$

There are just two agents with respect to this environment, which we shall refer to as Ag_1 and Ag_2 :

$$Ag_1(e_0) = \alpha_0$$

$$Ag_2(e_0) = \alpha_1$$

Assume the probabilities of the various runs are as follows:

$$P(e_0 \xrightarrow{\alpha_0} e_1 \mid Ag_1, Env_1) = 0.2$$

$$P(e_0 \xrightarrow{\alpha_0} e_2 \mid Ag_1, Env_1) = 0.2$$

$$P(e_0 \xrightarrow{\alpha_0} e_3 \mid Ag_1, Env_1) = 0.6$$

$$P(e_0 \xrightarrow{\alpha_1} e_4 \mid Ag_2, Env_1) = 0.2$$

$$P(e_0 \xrightarrow{\alpha_1} e_5 \mid Ag_2, Env_1) = 0.3$$

$$P(e_0 \xrightarrow{\alpha_1} e_6 \mid Ag_2, Env_1) = 0.5$$

Assume the utility function u_1 is defined as follows:

$$u_1(e_0 \xrightarrow{\alpha_0} e_1) = 8$$

$$u_1(e_0 \xrightarrow{\alpha_0} e_2) = 7$$

$$u_1(e_0 \xrightarrow{\alpha_0} e_3) = 4$$

$$u_1(e_0 \xrightarrow{\alpha_1} e_4) = 8$$

$$u_1(e_0 \xrightarrow{\alpha_1} e_5) = 2$$

$$u_1(e_0 \xrightarrow{\alpha_1} e_6) = 5$$

Given these definitions, determine the expected utility of the agents Ag_1 and Ag_2 with respect to Env_1 and u_1 , and explain which agent is optimal with respect to Env_1 and u_1 . Include an explanation of your calculations in your solution. **(13 marks)**

Q1.c answer

Given the utility function u_1 in the question, we have two transition functions, defined as $\tau(e_0 \xrightarrow{\alpha_0}) = \{e_1, e_2, e_3\}$, and $\tau(e_0 \xrightarrow{\alpha_1}) = \{e_4, e_5, e_6\}$. The probabilities of the various runs (three for each agent) is given in the question, along with the probability of each run occurring. Given the definition of the utility function u_1 , the *expected utilities* of agents Ag_0 and Ag_1 in environment Env can be calculated using:

$$EU(Ag, Env) = \sum_{r \in \mathcal{R}(Ag, Env)} u(r)P(r | Ag, Env).$$

This is equivalent to calculating the sum of the product of each utility for a run ending in some state with the probability of performing that run; i.e.

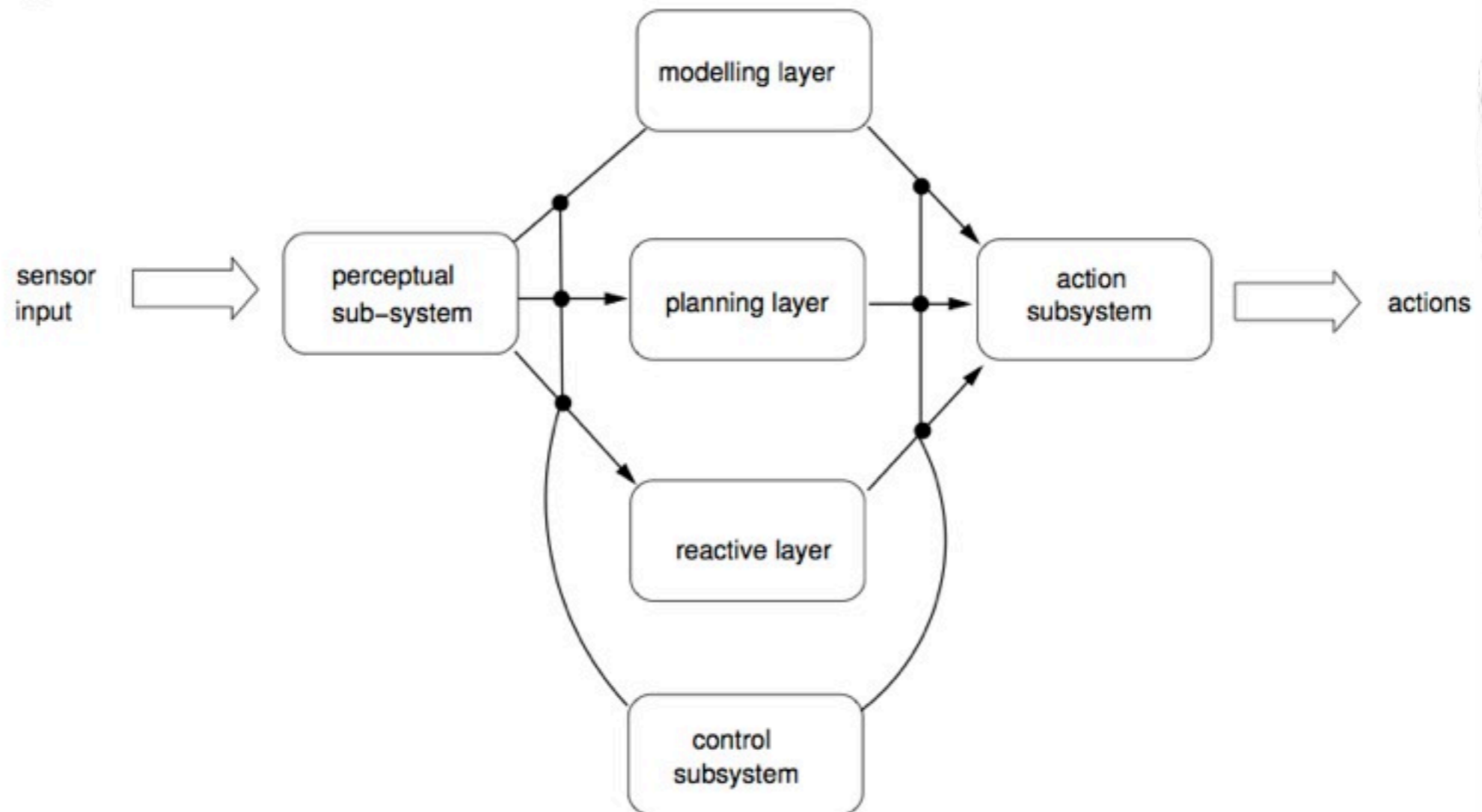
- Utility of $Ag_0 = (0.2 \times 8) + (0.2 \times 7) + (0.6 \times 4) = 1.6 + 1.4 + 2.4 = 5.4$
- Utility of $Ag_1 = (0.2 \times 8) + (0.3 \times 2) + (0.5 \times 5) = 1.6 + 0.6 + 2.5 = 4.7$

Therefore agent Ag_0 is optimal.

This is discussed further in Chapter 2. Ensure that all working, and also the equations used are stated and explained.

Question 2.a

2. (a) The following diagram illustrates the key subsystems of the TOURINGMACHINES agent architecture:



Describe the overall operation of the architecture, making sure you explain how the three decision layers achieve the goal of reactive, pro-active behaviour. **(15 marks)**

Q2.a answer

The Turing Machines architecture is an example of a hybrid architecture that combines reactive behaviour with that of deliberative, or pro-active behaviour. It consists of three layers, each of which operate in parallel. Each has access to the perceptual sub-system, which is responsible for converting the percepts obtained from sensor input into predicates that can be used for reasoning. In addition, each layer can result in the generation of actions that can then be executed. A control subsystem is responsible for monitoring the incoming percepts held in the perceptual sub-system, and then determining which of the actions (if any) should be executed from the different layers; i.e. it determines which layer is responsible for controlling the agent. In particular, the control subsystem can suppress sensor information going to certain layers, or it can censor actions generated by the different layers.

The reactive layer is responsible for responding to changes in the environment (in a similar way to Brooks subsumption architecture). A set of situation-action rules are defined, which then fire if they map to sensor input. For example, if the agent is controlling an autonomous vehicle and it detects a kerb unexpectedly in front of the vehicle, it can stop (or slow down) and turn to avoid the kerb.

The planning layer is responsible for determining the actions necessary to achieve the agent's goals. Under normal operation, this layer determines what the agent should do. This is done by making use of a set of planning schema, relating to different goals, and then performing the necessary actions. Note that no low level planning is performed.

The modelling layer represents the various entities in the world. This is responsible for modelling the world, including other agents, and for determining the agents goals, or planning goals that resolve any conflicts with other agents if such conflicts are detected. Whenever a goal is generated, it is passed onto the planning layer, which then determines the final actions.

Although several of the details here are from your notes, much more description was originally given in the lecture, and is also available from the course text book.

Question 2.b

(b) The Blocksworld scenario is represented by an ontology with the following formulae:

$On(x, y)$ obj x on top of obj y
 $OnTable(x)$ obj x is on the table
 $Clear(x)$ nothing is on top of obj x
 $Holding(x)$ arm is holding x

An agent has a set of actions Ac , such that $Ac = \{Stack, UnStack, Pickup, PutDown\}$:

$Stack(x, y)$
pre $Clear(y) \ \& \ Holding(x)$
del $Clear(y) \ \& \ Holding(x)$
add $ArmEmpty \ \& \ On(x, y)$
$UnStack(x, y)$
pre $On(x, y) \ \& \ Clear(x) \ \& \ ArmEmpty$
del $On(x, y) \ \& \ ArmEmpty$
add $Holding(x) \ \& \ Clear(y)$
$Pickup(x)$
pre $Clear(x) \ \& \ OnTable(x) \ \& \ ArmEmpty$
del $OnTable(x) \ \& \ ArmEmpty$
add $Holding(x)$
$PutDown(x)$
pre $Holding(x)$
del $Holding(x)$
add $OnTable(x) \ \& \ ArmEmpty \ \& \ Clear(x)$

It also has the following beliefs B_0 regarding the three bricks $\{A, B, C\}$, and the intention i :

$Beliefs B_0$	$Intention i$
$Clear(B)$	$Clear(A)$
$Clear(C)$	$Clear(B)$
$On(C, A)$	$On(B, C)$
$OnTable(A)$	$OnTable(A)$
$OnTable(B)$	$OnTable(C)$
$ArmEmpty$	$ArmEmpty$

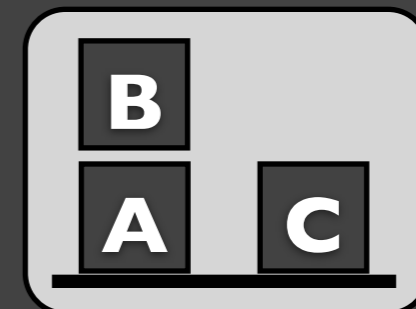
Calculate a plan π that would achieve i , given the beliefs B_0 . Draw the environment at the beginning of the plan, and after every time a $Stack$ or $UnStack$ action is performed. (10 marks)

Q2.b answer

A plan π is a sequence of actions, where each action results in changing the set of beliefs the agent has, until the final set of beliefs matches that of the intentions, such that $B_0 \xrightarrow{\alpha_1} B_1 \xrightarrow{\alpha_2} \dots \xrightarrow{\alpha_n} B_n$. Therefore, a planner will explore all the different possible sequences of actions to determine which one will result in the final set of intentions.

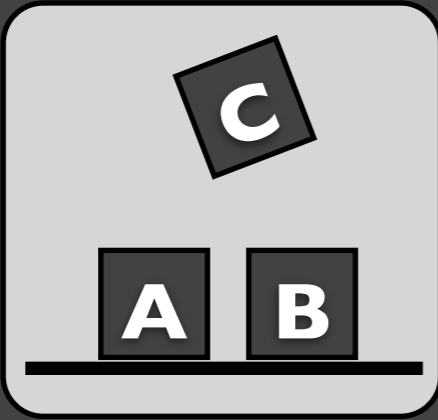
In this solution, only those actions that result in the final solution are given, with the set of beliefs that result in each step presented. The aim is to start with an initial set of beliefs, B_0 , and arrive at a final set of beliefs, B_n which corresponds to the intentions given in the question - i.e.

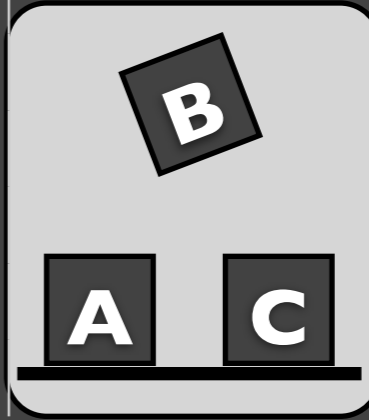
<i>Beliefs B_0</i>	<i>Intention i</i>
<i>Clear(B)</i>	<i>Clear(A)</i>
<i>Clear(C)</i>	<i>Clear(B)</i>
<i>On(C, A)</i>	<i>On(B, C)</i>
<i>OnTable(A)</i>	<i>OnTable(A)</i>
<i>OnTable(B)</i>	<i>OnTable(C)</i>
<i>ArmEmpty</i>	<i>ArmEmpty</i>

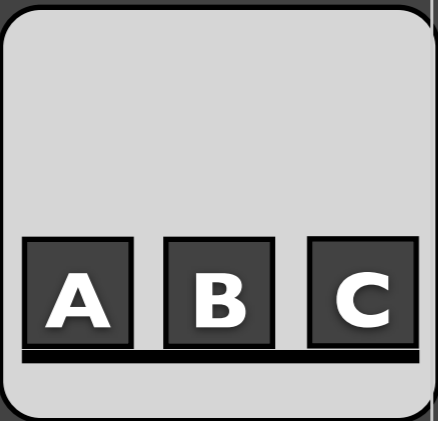


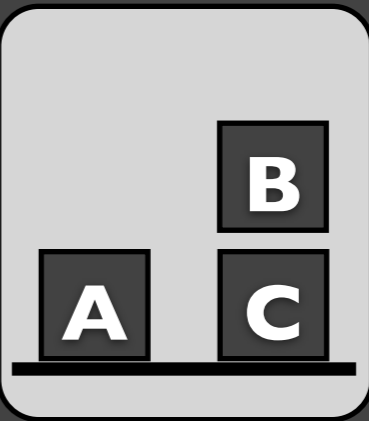
The solution is given on the next slide. In each case, the beliefs that hold prior to the action are given in bold, and the beliefs that are new after the action are also presented in bold.

Q2.b answer

<i>Beliefs B₀</i>	<i>Action</i>	<i>Beliefs B₁</i>
<i>Clear(B)</i> Clear(C) On(C, A) <i>OnTable(A)</i> <i>OnTable(B)</i> ArmEmpty	<i>Unstack(C, A)</i> 	<i>Clear(B)</i> <i>Clear(C)</i> <i>On(C, A)</i> <i>OnTable(A)</i> <i>OnTable(B)</i> <i>ArmEmpty</i> Holding(C) Clear(A)

<i>Beliefs B₂</i>	<i>Action</i>	<i>Beliefs B₃</i>
Clear(B) <i>Clear(C)</i> <i>OnTable(A)</i> OnTable(B) <i>Clear(A)</i> <i>OnTable(C)</i> ArmEmpty	<i>Pickup(B)</i> 	<i>Clear(B)</i> <i>Clear(C)</i> <i>OnTable(A)</i> <i>OnTable(B)</i> <i>Clear(A)</i> <i>OnTable(C)</i> <i>ArmEmpty</i> Holding(B)

<i>Beliefs B₁</i>	<i>Action</i>	<i>Beliefs B₂</i>
<i>Clear(B)</i> <i>Clear(C)</i> <i>OnTable(A)</i> <i>OnTable(B)</i> Holding(C) <i>Clear(A)</i>	<i>PutDown(C)</i> 	<i>Clear(B)</i> Clear(C) <i>OnTable(A)</i> <i>OnTable(B)</i> <i>Holding(C)</i> <i>Clear(A)</i> OnTable(C) ArmEmpty

<i>Beliefs B₃</i>	<i>Action</i>	<i>Beliefs B₄</i>
<i>Clear(B)</i> Clear(C) <i>OnTable(A)</i> <i>Clear(A)</i> <i>OnTable(C)</i> Holding(B)	<i>Stack(B, C)</i> 	<i>Clear(B)</i> <i>Clear(C)</i> <i>OnTable(A)</i> <i>Clear(A)</i> <i>OnTable(C)</i> <i>Holding(B)</i> ArmEmpty On(B, C)

The beliefs B₄, once rearranged, are now equivalent to the intentions.

Q3.a answer

3. (a) A *coalitional game with transferable payoff* is a structure $\langle Ag, \nu \rangle$. Explain what each of the components in the structure are, what they are intended to represent, and what the computational issues are if such structures are to be used for reasoning about multi-agent systems. **(5 marks)**

A coalitional game with transferrable payoff is represented by a tuple, consisting of Ag which is the set of agents that could appear in the coalition, and the characteristic function of the game, which assigns to every single combination of coalition a numeric value corresponding to the payoff that coalition may get. Thus if a coalition consisting of a specific set of agents, the characteristic function would include a real value corresponding to the payoff that the agents (as a coalition) should receive. The characteristic function does not state how the agents should distribute the payoff amongst themselves.

The problem with characteristic functions is with respect to representing them, as the number of payoff values and coalition structures is exponential to the size of the set of agents. Thus, processing characteristic functions for large sets of agents is possibly untenable. Therefore two approaches are adopted to represent characteristic functions; either try to find a complete representation that is succinct in most cases, or try to find a representation that is not complete, but that is always succinct.

Again, the above solution possibly contains more information than is strictly necessary. The important points to address is the notion of the characteristic function, the set of agents, and that simple approaches to representing characteristic functions can be exponential in size.

Question 3b

(b) In the context of cooperative games, consider the following marginal contribution net:

$$a \wedge b \rightarrow 6$$

$$b \rightarrow 3$$

$$c \rightarrow 4$$

$$b \wedge \neg c \rightarrow 2$$

Let ν be the characteristic function defined by these rules. Give the values of the following:

- (i) $\nu(\{a\})$
- (ii) $\nu(\{c\})$
- (iii) $\nu(\{a, b\})$
- (iv) $\nu(\{b, c\})$
- (v) $\nu(\{a, b, c\})$

(10 marks)

Q3.b answer

$$v(\{a\}) = 0$$

no rules apply

$$v(\{c\}) = 4$$

3rd rule

$$v(\{a, b\}) = 6 + 3 + 2 = 11$$

1st, 2nd and 4th rules

$$v(\{b, c\}) = 3 + 4 = 7$$

2nd and 3rd rule

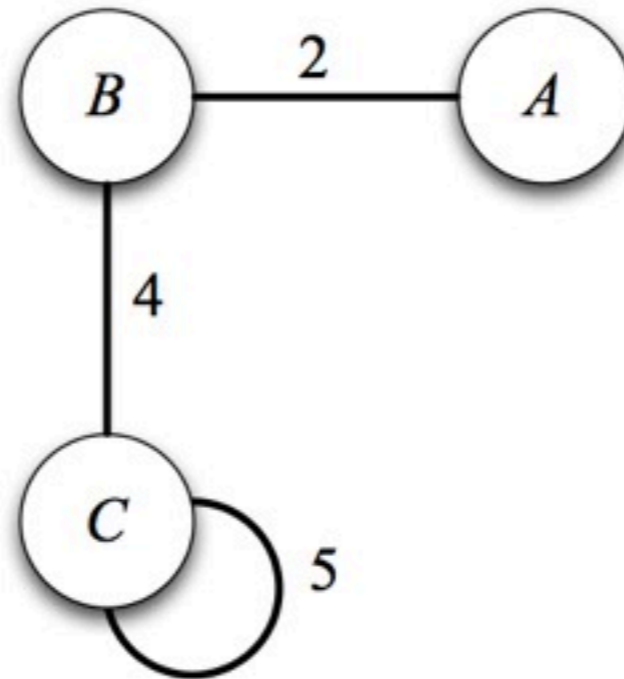
$$v(\{a, b, c\}) = 6 + 3 + 4 = 13$$

1st, 2nd and 3rd rules

The above solution would be sufficient, although include a note as to what rules fire. It would be difficult to give full marks to each solution if only the values were given, and there was no clarity as to where the results came from.

Question 3.c

c) Consider the following weighted subgraph representation of a characteristic function:



Let ν be the characteristic function defined by this graph.

- (i) Give the values of $\nu(\{A, B\})$, $\nu(\{C\})$, and $\nu(\{A, B, C\})$. **(6 marks)**
- (ii) Give an example of a payoff distribution that is in the core of this game, and an example of a payoff distribution that is not in the core of this game. In both cases, justify your answer. **(4 marks)**

Q3.c answer

i) The values are as follows:

$v(\{A,B\}) = 2$ - This corresponds to the single arc between the nodes a and b .

$v(\{C\}) = 5$ - The node c in the graph has a single self-referencing arc, with the weight 5.

$v(\{A,B,C\}) = 2+4+5 = 11$ - In this case we consider all of the arcs between nodes a , b , and c , and the single self referencing arc for c .

ii) The payoff distribution $\{2, 3, 6\}$ for the coalition $\{A, B, C\}$ would be in the core of the game, as A would get as least as much in this coalition as any other coalition. C can get at least 5 if it creates a coalition containing only itself, so it needs to receive at least 5. In this case it gets 6. B could try to form a coalition with C , but would only get to share a payoff of 4 with C getting 5. However, this would be rejected, as C could get more from joining the proposed coalition. B could get at most 2 if it formed a coalition with A . Therefore the payoff of 3 is better than it could get from any other coalition.

Conversely, a payoff distribution $\{8,2,1\}$ would not be in the core, as C would defect to create a coalition on its own, and gain a payoff of 5.

Although the explanation for the results in part i are not strictly necessary, they help clarify the answers. Each answer would get 2 marks each. Part ii is a little more challenging, but remember that part i defines some of the characteristic function, and that agents don't always have to work together. There is no single correct answer for this part of the question.

Question 4a and b

4. (a) In a combinatorial auction, a bid is made for the following bundles:

$$B_i = (\{a, b\}, 4) \text{ XOR } (\{c, d\}, 7)$$

Calculate what would be paid for the following bundles:

- (i) $v_{\beta_1}(\{a\})$
- (ii) $v_{\beta_1}(\{a, b\})$
- (iii) $v_{\beta_1}(\{a, b, c\})$
- (iv) $v_{\beta_1}(\{a, b, c, d\})$

(8 marks)

- (b) Describe the *Vickrey Auction*, and explain why it is *incentive compatible*; i.e. it is in the bidder's interest to bid their true value.

(5 marks)

Q4.a & b answer

a) The solution is as follows:

$v_{\beta_1}(\{a\}) = 0$ No bid value is given for a bundle for the single value a

$v_{\beta_1}(\{a, b\}) = 4$ The bid value for a bundle for $\{a, b\}$ is 4

$v_{\beta_1}(\{a, b, c\}) = 4$ Although c is in the bundle, the only matching bid is $\{a, b\}$

$v_{\beta_1}(\{a, b, c, d\}) = 7$ Both bundles $\{a, b\}$ and $\{c, d\}$ match, so the largest bid is paid

b) The Vickrey auction is a second price sealed bid auction. Bidders submit their bids directly to the auctioneer. The winner is the bidder who bids the most. However, they then pay the second highest bid. This has the advantage of being incentive compatible, i.e. it encourages bidders to only bid their true valuation of the good. This is because:

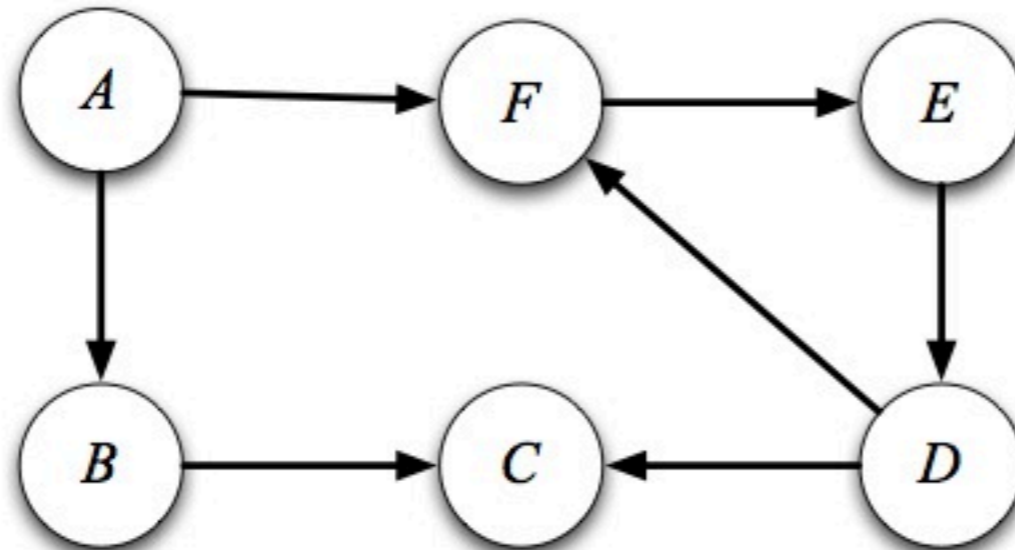
- if the bidder bids more than their true valuation, then they might win, but if another bidder bids more than the first bidder's true valuation, then the first bidder will pay more than they think the good is worth.

- if the bidder bids less than their true valuation, they still pay the second highest bid. However, the bidder stands less chance of winning the good.

Again, in part a - if it is clear that there is more than a single mark for a numerical answer, then give some explanation as to why you arrived at that answer!!!

Question 4.c

c) The following figure shows an abstract argumentation system.



Compute the grounded extension of this argument system, giving the status (*in* or *out*) of all of the six arguments in the graph, and explain why they are either *in* or *out*.

(2 marks each, for a total of 12 marks)

Q4.c answer

c) Nodes in the graph represent arguments, and arcs indicate the direction of attack from one argument to another.

A is in as no other argument attacks it.

B and F are out as they are attacked by A.

E is in, as it can only be attacked by F, but F is out.

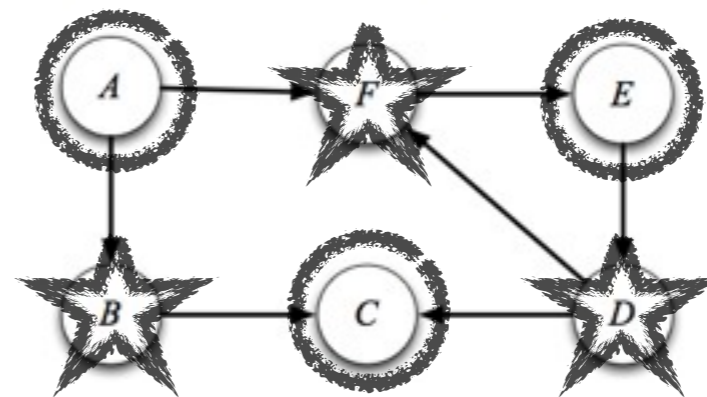
D is out, as it is attacked by E.

C could be attacked by either B or D. But as both B and D are out, it must be in

Therefore the Grounded Extension includes only those arguments that are left (which in this case is none) and those argument that are in; i.e {A, C, E}.

This answer is comparatively simple, but again, as there are more than one mark for each argument, ensure that the explanation is given. Don't be afraid of making notes on the exam script (see opposite).

The following graph shows an abstract argumentation system.



Compute the grounded extension of this argument system, giving the status (*in* or *out*) of all of the six arguments in the graph, and explain why they are either *in* or *out*.

(2 marks each, for a total of 12 marks)

Question 5a

5. The following payoff matrix (A) is for the “prisoner’s dilemma”:

		<i>i</i>	
		defect	coop
<i>j</i>	defect	2, 1	4, 1
	coop	4, 3	1, 3

The following payoff matrix (B) is for the “stag hunt”:

		<i>i</i>	
		defect	coop
<i>j</i>	defect	2, 3	1, 3
	coop	3, 4	1, 4

The following payoff matrix (C) is for “matching pennies”:

		<i>i</i>	
		defect	coop
<i>j</i>	defect	-1, 1	1, -1
	coop	1, -1	-1, 1

(a) For each of these payoff matrices:

- (i) Identify all (pure strategy) Nash Equilibria;
- (i) Identify all Pareto optimal outcomes;
- (iii) Identify all outcomes that maximise social welfare.

(18 marks)

Q5.a answer

i) Pure Strategy Nash Equilibria

A pure strategy Nash Equilibria is one where, given two agents A and B, A can do no better than its NE strategy when it knows B's strategy, and vice versa

A) (Prisoner's Dilemma): DD

B) (Stag Hunt): DD and CC

C) (Matching Pennies): no pure NE strategies

ii) Pareto Optimal Outcomes

A Pareto Optimal Outcome is one where no agent can change its strategy without adversely affecting the other.

A) (Prisoner's Dilemma): DC, CD, CC (the payoff for DD can be improved for both agents by both cooperating)

B) (Stag Hunt): CC (in every other case, both payoffs can be improved by cooperating)

C) (Matching Pennies): none

iii) Maximise social Welfare

This is the strategy that optimises the global payoff for both agents

A) (Prisoner's Dilemma): CC (payoff is 6)

B) (Stag Hunt): CC (payoff is 8)

C) (Matching Pennies): none (or all, as the payoff in each case is zero)

Although in each case, these can simply be memorised, it is better to try to understand how to calculate these, as an exam question might also include some non-standard payoff matrix.

Q5.b

(b) “Program equilibria make cooperation possible in the one-shot prisoner’s dilemma”.
Explain and critically assess this statement. (7 marks)

One of the main problems with the one-shot prisoner's dilemma, is that although it would be better for both agents to cooperate, it is only in one agent's interest to cooperate if the other agent is cooperating. However, if an agent is forewarned that the other agent will cooperate, then an agent can improve its payoff by defecting. The fact that the game is a one shot game means that an agent does not care about retaliation!

Program equilibria makes cooperation possible by removing the fact that an agent can be forewarned about the other agent's intention. Both agents submit a program to a mediator, which describes the agents strategies and conditions for voting. In particular, the programs can determine how one agent should vote depending on the other agent's program. The programs can only be seen by the mediator, and the mediator then runs these programs to determine how each agent should behave. Thus, an agent can submit a program that states that if the other agent's program is the same then the agent should cooperate, otherwise it should defect. This way, an agent can state that it is only prepared to cooperate if and only if the second agent will also cooperate, otherwise it will defect.

This is simply a matter of understanding the process.