PAPER CODE NO. **COMP310**	EXAMINER : Michael Wooldridge DEPARTMENT : Computer Science Tel. No. 7790

UNIVERSITY OF LIVERPOOL

MAY 2008 EXAMINATIONS

Multiagent Systems

TIME ALLOWED : Two and a Half hours

INSTRUCTIONS TO CANDIDATES

Answer **four** questions.

If you attempt to answer more questions than the required number of questions (in any section), the marks awarded for the excess questions answered will be discarded (starting with your lowest mark).

Question 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) The following pseudo-code characterises the action selection process of a logic-based agent.

for each $\alpha \in Ac$ do
    if $\Delta \vdash_\rho Do(\alpha)$ then
        return α
    end-if
end-for
for each $\alpha \in Ac$ do
    if $\Delta \not\vdash_\rho \neg Do(\alpha)$ then
        return α
    end-if
end-for
return `null`

Describe how this loop works, explaining in particular the role of ρ, Δ, $\vdash_\rho$, and $Do(\alpha)$, and the limitations of such an approach for run-time decision-making.

[12 marks]

c) With particular reference to the outline architecture above, explain what you understand by the term *calculative rationality*.

[4 marks]

d) Explain what you understand by *bounded optimality*.

[5 marks]

Question 2

The following pseudo-code defines a control loop for a practical reasoning ("BDI") agent.

```
B := B_0;
I := I_0;
while true do
    get next percept ρ;
    B := brf(B, ρ);
    D := options(B, I);
    I := filter(B, D, I);
    π := plan(B, I);
    while not (empty(π) or succeeded(I, B) or impossible(I, B)) do
        α := hd(π);
        execute(α);
        π := tail(π);
        get next percept ρ;
        B := brf(B, ρ);
        if reconsider(I, B) then
            D := options(B, I);
            I := filter(B, D, I);
        end-if
        if not sound(π, I, B) then
            π := plan(B, I)
        end-if
    end-while
end-while
```

a) Explain the overall working of this loop, with particular reference to the following components:

B_0, B
D
I_0, I
$\mathit{brf}(\cdots)$
$\mathit{options}(\cdots)$
$\mathit{filter}(\cdots)$
$\mathit{plan}(\cdots), \pi, \mathit{hd}(\cdots), \mathit{tail}(\cdots)$
$\mathit{succeeded}(\cdots)$
$\mathit{impossible}(\cdots)$
$\mathit{sound}(\cdots)$

[20 marks]

b) "A rational agent is committed to intentions as long as it believes these are achievable and have not been achieved." Describe how the control loop above satisfies this criterion.

[5 marks]

Question 3

a) Consider the coalitional game with agents $Ag = \{1, 2\}$ and characteristic function v defined by $\nu(\{1\}) = 5$, $\nu(\{2\}) = 5$, $\nu(\{1, 2\}) = 20$. With reference to this example, and with the aid of examples, explain what you understand by a coalitional game and what you understand by the *core* of a coalitional game.

[5 marks]

b) The *Shapley value* attempts to provide an answer to the question of how to fairly divide coalitional value amongst coalition members. The Shapley value is the unique solution to a set of axioms characterising fair division. Explain and if possible define the Shapley value and the axioms that characterise it, with reference to the coalitional game introduced in part (a) above.

(A pass mark in this question may be obtained with an informal answer, but full marks can only be obtained with the formal definition of the Shapley value.)

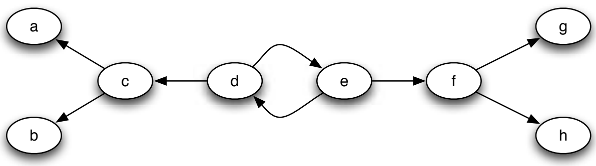
[10 marks]

c) Coalitional games present several problems from a computational perspective. Explain what you understand these problems to be, and give *two* examples of representations of coalitional games that are intended to overcome these problems.

[10 marks]

Question 4

The following figure shows an abstract argument system.



a) Give an example of a non-empty preferred extension for this argument system, if one exists, else justify why none exists.

[2 marks]

b) Give pseudo-code to describe how to compute the *grounded extension* of an abstract argument system.

[5 marks]

c) With reference to the abstract argument system above, compute the *grounded extension*, giving the ultimate status (in or out) of all arguments. Now suppose the edge from *d* to *e* is deleted (but *not* the edge from *e* to *d*): compute the grounded extension in this case also.

[18 marks]

UNIVERSITY OF LIVERPOOL

Question 5

Consider the following payoff matrix, for the "prisoner's dilemma".

		i	
		defect	coop
j	defect	2 2	1 4
	coop	4 1	3 3

With reference to this payoff matrix:

a) Define the notion of a Nash equilibrium strategy pair, explain why the concept of Nash equilibrium strategies is so important, and identify with justification which (if any) strategy pairs in this payoff matrix are in Nash equilibrium.

[5 marks]

b) Define the notion of a Pareto optimal outcome, explain why the concept of Pareto optimal outcomes is so important, and identify with justification which (if any) outcomes in this payoff matrix are Pareto optimal.

[5 marks]

c) Suppose two agents are negotiating. With reference to the notion of Nash equilibrium that you defined in (a), above, and with the aid of an example, justify why it can sometimes be beneficial for an agent to truthfully reveal its negotiation strategy to other negotiation participants.

[5 marks]

d) The prisoner's dilemma is often interpreted as being proof that somehow cooperation between self interested agents is impossible. Explain why this is, and discuss the circumstances under which cooperation seems to be possible in the prisoner's dilemma, with reference to Axelrod's tournament.

[10 marks]