E3.16 Artificial Intelligence

There are SIX questions. *Answer FOUR questions*.

Examiner Responsible Dr J. V. Pitt

First Marker	**Dr J. V. Pitt**	**.....**	**.....**
Second Marker	**Dr M. P. Shanahan**	**.....**	**.....**

Special Information for Invigilators: NIL

Information for Candidates:

The Prolog General Graph Search Program (GGS)

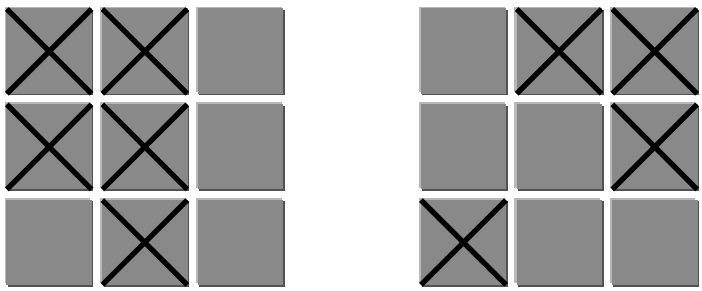
```
/* search( +Paths, ?Path ) succeeds when
Path is an extension of some path in Paths to a goal
*/

search( Paths, [Node|Path] ) :-
     choose( [Node|Path], Paths, _ ),
     state_of( Node, State ),
     goal_state( State ).
search( Paths, SolnPath ) :-
     choose( Path, Paths, OtherPaths ),
     one_step_extensions( Path, NewPaths ),
     add_to_paths( NewPaths, OtherPaths, AllPaths ),
     search( AllPaths, SolnPath ).
```

KE Inference Rules for Modal Logic S5 using Fitting's prefixed tableau

$\underline{i\colon \Box p}$ $j\colon p$	$\underline{i\colon \neg\Diamond p}$ $j\colon \neg p$	*for any available integer j*
$\underline{i\colon \neg\Box p}$ $j\colon \neg p$	$\underline{i\colon \Diamond p}$ $j\colon p$	*for j an integer new to the branch*

Note: *i* and *j* are integers labelling possible worlds.

1. The game of Magic Checkboxes is played with two 3-by-3 squares as shown below.



The object of the game is to match the checkboxes in the left hand square to the checkboxes in the right hand square. The catch (there is always a catch) is that clicking a checkbox in the left hand square has the following results:

- a click on a corner box toggles the state of the box and all three of its neighbours;
- a click on a box in the middle of a side toggles the state of only the three boxes on that side;
- a click on the centre box toggles the state of the five boxes that form a cross (up and down, not diagonally) in the middle of the square.

(a) Formulate a representation of the search space for this problem using a form of declarative specification (not necessarily Prolog). Specify state change rules for clicking in the top right corner, centre, and bottom middle boxes only.

[8]

(b) Compare and contrast *breadth first* and *depth first* search strategies with respect to their completeness, complexity and optimality.
Use this, and any other criteria, to decide which strategy would be preferable for solving the Magic Checkbox game.

[3]

(c) You are told that from any starting configuration of left hand square, to get to any configuration of the right hand square, you would only need to select any box just once.
How would this affect your choice of search strategy in part (b), and explain briefly how it could be incorporated in the General Graph Search Program?

[3]

(d) Describe briefly *iterative deepening depth first* search.
Briefly describe how the General Graph Search program needs to be modified to perform iterative deepening depth first search.
Comment on its completeness, complexity and optimality with respect to breadth first and depth search. Give reasons for your answer.

[6]

2. (a) Describe briefly *uniform cost* search.
Indicate how a search space needs to be represented in order to perform uniform cost search, and indicate how the General Graph Search program needs to be modified.

[3]

(b) Describe briefly *best first* search.
Compare and contrast best first search with uniform cost search.

[3]

(c) Describe the *A* algorithm* and its relationship to best first and uniform cost search.

[3]

(d) Define what is meant by an *admissible heuristic.*
Why is admissibility important in A* search?
For path finding in grid mazes, explain why both straight line distance and Manhattan distance are admissible heuristics.

[3]

(e) Define what is meant by a *monotonic function*.
Give the *pathmax equation* and explain how it is used in A* search.
For path finding in grid mazes, explain why the total (estimated) path cost function in A* search is monotonic.

[3]

(f) Explain why, even though there may be several solution paths in a grid maze, the first solution found by A* search must be optimal.

[5]

3. (a) What is meant by a *conceptualization* in Knowledge Representation.
Give different ways in which colour could be conceptualised, for example in a blocks world with differently coloured blocks.

[3]

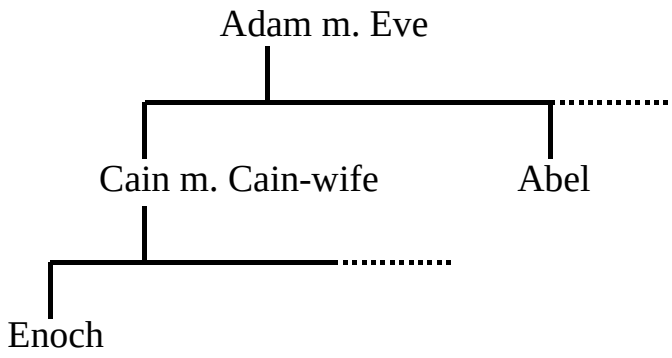
(b) Define what is meant by *unification* in logic programming.
Briefly describe a unification algorithm for two terms.

[4]

(c) Describe the inference rule *resolution*, and show that it is sound.

[4]

(d) Consider the fragment of a family tree shown below:



(i) Define the facts shown using mother, father and married relations only;

(ii) Define rules for the parent and grandfather relations;

(iii) Show the facts and the rules of parts d(i) and d(ii) as horn clauses (with the rules implicitly universally quantified);

(iv) Using unification and resolution, show that Adam is the grandfather of Enoch.

[9]

4. (a) In *Asterix and the Great Crossing*, while Obelix has gone hunting, Asterix is captured by some Native Americans. On returning Obelix finds only Asterix's helmet. Obelix, realising that Asterix is not there and has taken his helmet off, then reasons as follows:

"Asterix only takes his helmet off if he is eating or sleeping. He's not eating [because he's waiting for me to come back from hunting]. He's not sleeping, or he'd be here. So, something must have happened..."

(i) Formalise Obelix's knowledge and thoughts literally as statements of propositional logic. Show which are premises and what is his conclusion.

(ii) Taking just the set of premises of part a(i), use the **KE** proof procedure build a model and show that it is inconsistent. What is the implication of reasoning with an inconsistent set of premises?

(iii) How does Obelix need to change his premises so that his reasoning is sound? Illustrate your answer by building a model with an open branch that can only be closed by the (negation of the) conclusion.

[5]

(b) (i) What is the *subformula property*?
Why is it useful in an automated theorem prover based on the **KE** calculus?
How does it affect soundness and completeness of the **KE** proof procedure?

(ii) Given a **KE** proof of $\vdash \neg P \vee Q$ and a **KE** proof of $\vdash P$, prove that $\vdash Q$.
(NB: you are not asked to show $\{\neg P \vee Q, P\} \vdash Q$, and you do not have to behave like an automated theorem prover.)

[5]

(c) What is a (Kripke) model for a modal logic?
State how such a model is used to determine the meaning (truth value) of modal formulas like $\Box p$ and $\Diamond p$, and an ordinary propositional symbol (e.g. p).

[4]

(d) Using the **KE** proof procedure and Fitting's prefixed tableau, determine which of the following formulas are theorems of the modal logic S5:

(i) $\Box\Diamond p \leftrightarrow \Diamond p$
(ii) $\Box\Diamond p \leftrightarrow \Box p$
(iii) $\Box\Diamond p \leftrightarrow \Diamond\Box p$
(iv) $\Diamond\Box p \leftrightarrow \Diamond\Box p$

For formulas that are not theorems, give a Kripke model which provides a counter-example (i.e. a model which satisfies the negation of the formula).

[6]

5. (a) Define a set of attributes, some of which might be exhibited by an embedded software process, which would justify calling that process an *intelligent agent*.

[5]

(b) Consider the following user agent operating in a distributed multi-agent system:

The user agent is a personal service agent; i.e. an agent which takes a user interest (or interests) stored in a user profile, for example, and provides information on web sites which match that interest, or retail outlets offering special deals, or other people with the same interest.

State, with reasons or examples, which attributes defined in part (a) might be required of such an agent.

[5]

(c) Explain, using an example, why the BDI (Beliefs-Desires-Intentions) model is useful for describing the behavioural requirements of an intelligent agent.

[5]

(d) Draw a practical BDI architecture for a 'rational agent'. State the principal function of each component, and briefly indicate how they interact with each other during one cycle of the interpreter execution cycle.

[5]

6. (a) Describe briefly *KQML* (Knowledge Query Manipulation Language), indicating its purpose and conceptual layers, and the syntactic form and semantic model of KQML messages.

[5]

(b) What is the role of a *Facilitator* agent in KQML?
Show the sequence of KQML messages (i.e. the protocol) exchanged between two agents *A* and *B* and a facilitator *F* in which:

(i) *A* asks *F* to "subscribe" to a service provided by another agent (*B*, say)

(ii) *A* asks *F* to "recommend" an agent (*B*, say) capable of providing a service.

Briefly comment on whether or not these protocols are complete.

[5]

(c) Considering a BDI (Beliefs-Desires-Intentions) agent, give a logical formulation and English description of:

(i) axioms combining the beliefs and desires of a BDI agent that trigger intentions to perform *inform*, *query*, and *command* speech acts (so that here the agent is the *sender*);

(ii) axioms for the change in belief state of a BDI agent which 'observes' an *inform*, *query* and *command* speech act (so that here the agent is the *receiver*).

[6]

(d) Briefly comment on the notions of sincerity, trust, authority and belief revision as they affect the logical formulations in part (b).

[4]

E3.16 Artificial Intelligence

Examiner **Dr J. V. Pitt**

First Marker **Dr J. V. Pitt**

Second Marker **Dr M. P. Shanahan**

There are SIX Questions. Answer FOUR.

MODEL ANSWERS

ANSWER 1

MARKING SCHEME
(a) 8 marks
(b) 3 marks
(c) 3 marks
(d) 6 marks

(a)
State = 9-tuple (A,B,C,D,E,F,G,H,I)
Each argument represents one box
3x3 flattened into 9x1 row by row
each variable takes vale on/off according to if box is checked or not

Start State = (on,on,off,on,on,off,off,on,off)

Goal state = (off,on,on,off,off,on,on,off,off)

```
state_change( topright,
        (A1,B1,C1,D1,E1,F1,G1,H1,I1), (A2,B2,C1,D2,E2,F1,G1,H1,I1) :-
                opposite( A1, A2 ), opposite( B1, B2 ),
                opposite( C1, C2 ), opposite( D1, D2 ).

state_change( centre,
        (A1,B1,C1,D1,E1,F1,G1,H1,I1), (A1,B2,C1,D2,E2,F2,G1,H2,I1) :-
                opposite( B1, B2 ), opposite( D1, D2 ),
                opposite( E1, E2 ), opposite( F1, F2 ), opposite( H1, H2 ).

state_change( bottommiddle,
        (A1,B1,C1,D1,E1,F1,G1,H1,I1), (A2,B2,C1,D2,E2,F1,G2,H2,I2) :-
                opposite( G1, G2 ), opposite( H1, H2 ), opposite( I1, I2 ).

opposite( on, off ).
opposite( off, on ).
```

(b)
Let m be the maximum depth of the tree, b be the average branching factor, and d be the depth at which a solution is found

	depth first	breadth first
completeness	no	yes
complexity – time	O(b^m)	O(b^d)
complexity – space	O(b*m)	O(b^d)
optimality	no	yes

Use breadth first for this problem, as search space is small even if branching factor is high and there is the possibility of loops.

(c):
put loop checking in

still use breadth first just prune out some useless paths

in the node representation, store the rules used to get to this node as well as the state
in one step extension, when it comes to make the new node, check that the rule to be used has not already occurred in this list

(d)
do depth first search at successive depths, depth-limited search at incremented depths, i.e. depth limit = 0 (just root node), depth limit = 1 (root plus its successors), depth limit = 2 (root plus its successors, and there successors)

idea is that at each iteration, we choose a path, if the frontier node is a goal state, we are done, otherwise, check the length of the path, if this = depth + 1, choose another path, otherwise expand the frontier node and extend the path, and recurse as normal

therefore changes to GGS:
put an extra clause in
put a wrapper for the iteration

id search combines the optimality and completeness of breadth first with the time and space complexity of breadth first

completeness: id search is complete because it effectively does exhaustive search like breadth first, but with depth first (limited) strategy

optimality: it will find optima solution provided optimal solution is one with shortest path from root node (initial state) to frontier node (goal state)

time complexity: O(b^d), even though mode nodes are expanded, the number expanded at deeper levels much greater than at higher levels, so worst case still *to the order of* O(b^d)

space complexity: O(b*d), only need b nodes on each level, but instead of maximum depth m reduced to depth of solution d

ANSWER 2

MARKING SCHEME
(a) 3 marks
(b) 3 marks
(c) 3 marks
(d) 3 marks
(e) 3 marks
(f) 5 marks

(a)
uniform cost search is modification of breadth first search: expand the lowest cost node on the search frontier, as measured by some path cost function *g*

search space modifications: nodes have to have at least one parameter to record the accumulated cost of getting here, as measured by *g*; state change rules have to include cost of applying rule;

modification to GGS: making a new node has to implement the path cost function; either insert the paths in order of increasing cost and take the head; or append new paths and search for cheapest when choosing which to expand next.

(b)
best first search expand the node on the search frontier with the least estimated cost of getting from that node to the goal state

	uniform cost	best first
optimality:	yes	no
complete	yes	no
space complexity	O(b^d)	O(b^m)
time	O(b^d)	O(b^m)

uniform complexity scores on optimality and completeness, complexity looks same order, but this is worst case: in practice time/space complexity much better with best first IF you have a good heuristic

(c)
A* search:
uniform cost search expands node with least cost from start node to current node using cost function *g*
best first search expands node with least estimated cost from current node to goal node using heuristic *h*
A* search sums *g* + *h* and expands node with least combined cost first

(d)
Admissible Heuristic: never over-estimates cost of getting from current to goal

Important because if we over-estimate we might not get optimal solution

both straight line distance and 'manhattan' distance are admissible because:
for straight line distance d_sl, in any right-angle triangle, d_sl < x + y, indeed d_sl is never greater than x +y, at best equal when there is a straight line.
therefore straight line distance heuristic never overestimates
for manhattan distance d_m, d_m = x + y,
so again never overestimates

(e)
A monotonic function is always either uniformly increasing or uniformly decreasing.

Pathmax equation: f(succ(n)) = max(f(n), g(succ(n)) + h(succ(n)))
Most admissible heuristics are monotonic, but those that are not can be forced to be by the pathmax equation

Path cost is monotonic because

$g(succ(n)) \geq g(n) + 1$ (move one grid square at a time)

$h(succ(n)) \geq h(n) - 1$ (at best we are one step closer, irrespective of heuristic)

so

$g(succ(n)) + h(succ(n)) \geq g(n) + h(n)$

path cost never decreases so it is monotonic

Note this assuming that there is no diagonal movement, but if there were, g could be distance travelled rather than grid units, and the reduction in h with either straight line distance or manhattan distance will be greater than equal to this, so the inequality still holds.

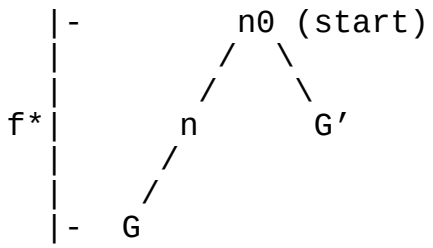
(f)
A* search returns optimal solution

Optimal solution has cost f* to get to optimal goal G
Suppose A* search returns path to sub-optimal goal G'
We show that this is impossible

$f(G') = g(G') + h(G')$
$= g(G') + 0$ G' is a goal state, we require h to be 0
$= g(G')$

If G' is sub-optimal then $g(G') > f^*$

Now consider a node n on path to optimal solution G



Then:	f*	≥	f(n)	monotonicity
	f(n)	≥	f(G')	otherwise A* expands n first
	f*	≥	f(G')	transitivity of [3]
	f*	≥	g(G')	a contradiction

So either G' was optimal or A* does not return a sub-optimal solution.

ANSWER 3

MARKING SCHEME
(a) 3 marks
(b) 4 marks
(c) 4 marks
(d) 9 marks

(a)
Conceptualisation = representation of knowledge in declarative form
formally 3-tuple <domain, functional basis, relational basis>

colour as element of domain	colour(block, red)
colour as function	colour(block) = red
colour as relation	red(block)

(b)
unification = process by is computed a set of substitutions (values for variables) that makes two terms the same

algorithm to unify two terms, X, and Y:
if X is a variable and Y is a variable, then they unify, substitute for each other
if X is a term, and Y is a variable, then they unify, substitution is Y <- X
if X is a variable, and Y is a term, then they unify, substitution is X <- Y
if X and Y are simple terms (constants), then they unify if they are identical
if X and Y are compound terms, then they unify if:
    their functors are the same
    they have the same number of arguments
    each pairwise corresponding argument unfies

(c)
resolution: rule of inference: from $p \vee q$ and $\neg p \vee r$, infer $q \vee r$
resolving p and ¬p is a contradiction
more general case:

To show it is sound, draw up a truth table, and show that when the premises are true, so is the conclusion

(d) showing just the relevant facts:
(i)
$f(a,c)$
$f(c,e)$

(ii)
$f(X,Y) \rightarrow p(X,Y)$
$f(X,Z) \wedge p(Z,Y) \rightarrow gf(X,Y)$

(iii)
$f(a,c)$
$f(c,e)$
$p(X,Y) \vee \neg f(X,Y)$
$gf(X,Y) \vee \neg(f(X,Z) \wedge p(Z,Y)) = gf(X,Y) \vee \neg f(X,Z) \vee \neg p(Z,Y)$

(iv)
query is $\neg gf(a,e)$, by resolution and unification:

$\neg f(a,Z) \vee \neg p(Z,e)$	S *is* $X = a,\ Y = e$
$\neg p(c,e)$	S *is* $Z = c$
$\neg f(c,e)$	S *is* $X = c,\ Y = e$

contradiction
so $gf(a,e)$ must be true

ANSWER 4

MARKING SCHEME
(a) 5 marks
(b) 5 marks
(c) 4 marks
(d) 6 marks

(a)
Let *tho* = takes his helmet off, *e* = eating, *s* = sleeping, *h* = here, *smhh* = something must have happened, then the premises are lines 1-4, with closed KE tree showing inconsistent set as follows:

1	$tho \wedge \neg h$	
2	$tho \rightarrow e \vee s$	
3	$\neg e$	
4	$\neg s \vee h$	
5	*tho*	*a*, 1
6	$\neg h$	*a*, 1
7	$e \vee s$	*b*, 2, 5
8	*s*	*b*, 7, 3
9	*h*	*b*, 4, 8

close 6, 9

Starting with an inconsistent set you can prove anything.
Needs to change premise 2 to $tho \rightarrow e \vee s \vee smhh$.

Then build tree:

		1	$tho \wedge \neg h$		
		2	$tho \rightarrow e \vee s \vee smhh$		
		3	$\neg e$		
		4	$\neg s \vee h$		
		5	$\neg smhh$	negated conclusion	
		6	*tho*	*a*, 1	
		7	$\neg h$	*a*, 1	
		8	$e \vee s \vee smhh$		*b*, 2, 6
		9	$s \vee smhh$		*b*, 8, 3
10	*s*	*PB*	11	$\neg s$	*PB*
12	*h*	*b*, 4,10	13	*smhh*	*b*,11, 9

close 7, 12 *open branch*, *closed on* 13 *and* 5

(b)
(i)
Subformula property: only need to apply PB (0-premise rule) to formulas or subformulas which appear on a branch of a KE-tableau, and not arbitrary formulas.

Important because restriction ensures proof procedure is decision procedure for propositional logic (i.e. terminates with yes or no result)

Doesn't affect soundness and completeness

(ii)
A proof of $\vdash \neg P \vee Q$ means there is a closed KE-tableau for $\neg(\neg P \vee Q)$, call this T1
A proof of $\vdash P$ means there is a closed KE-tableau for $\neg P$, call this T2

We want to show $\vdash Q$, so we start from $\neg Q$, and apply PB *but do not obey sub-formula property* (i.e. do not behave like an automated theorem prover), and then we get

1		1: $\neg Q$	
		||	
2(*PB*)	$\neg P \vee Q$	$\neg(\neg P \vee Q)$	3(*PB*)
4(*b*,1,2)	$\neg P$	$T1$	
	$T2$	$\otimes$	
(5,7)	$\otimes$		

(c)
A model M is a 3-tuple (W,R,D) where
W is a non-empty set of possible worlds
R is the accessibility relation on worlds
D is a mapping from propositional symbols to subsets of W (denotations)

Then for a model M and α a world an element of W in M, truth of $\Box p$, $\Diamond p$, and p
$\models^{M,\alpha}\Box p :\leftrightarrow \forall\beta.\ \alpha R\beta \rightarrow \models^{M,\beta} p$
$\models^{M,\alpha}\Diamond p :\leftrightarrow \exists\beta.\ \alpha R\beta \rightarrow \models^{M,\beta} p$
$\models^{M,\alpha} p :\leftrightarrow \alpha \in D(p)$

(d)
(*i*)

1	1: $\neg(\Box\Diamond p \leftrightarrow \Diamond p)$		
	||		
2(*PB*)	1: $\Box\Diamond p$	1: $\neg\Box\Diamond p$	3(*PB*)
4($\leftrightarrow$,1,2)	1: $\neg\Diamond p$	1: $\Diamond p$	8($\leftrightarrow$,1,3)
5($\Box$,2)	1: $\Diamond p$	2: $\neg\Diamond p$	9($\Diamond$,3)
6($\Diamond$,4)	2: p	3: p	10($\Diamond$,8)
7($\Box$,5)	2: $\neg p$	3: $\neg p$	11($\Box$,9)
(5,7)	$\otimes$	$\otimes$	(10,11)

(*ii*)

1	1: $\neg(\Box\Diamond p \leftrightarrow \Box p)$		
	||		
2(*PB*)	1: $\Box\Diamond p$	1: $\neg\Box\Diamond p$	3(*PB*)
4($\leftrightarrow$,1,2)	1: $\neg\Box p$	1: $\Box p$	8($\leftrightarrow$,1,3)
5($\Box$,4)	1: $\Diamond p$	2: $\neg\Diamond p$	9($\Diamond$,3)
6($\Diamond$,5)	2: p	2: p	10($\Box$,8)
7($\Diamond$,2)	3: $\neg p$	2: $\neg p$	11($\Box$,9)
		$\otimes$	(9,11)

(*iii*) 1 1: $\neg(\Box\Diamond p \leftrightarrow \Diamond\Box p)$

||

2(*PB*)	1: $\Box\Diamond p$	1: $\neg\Box\Diamond p$	3(*PB*)
4($\leftrightarrow$,1,2)	1: $\neg\Diamond\Box p$	1: $\Diamond\Box p$	8($\leftrightarrow$,1,3)
5($\Box$,2)	1: $\Diamond p$	2: $\neg\Diamond p$	9($\Diamond$,3)
6($\Box$,4)	1: $\neg\Box p$	3: $\Box p$	10($\Diamond$,8)
7($\Diamond$,5)	2: p	3: $\neg p$	11($\Box$,9)
8($\Diamond$,6)	3: p	3: p	11($\Box$,10)
		$\otimes$	(9,11)

(*iv*) 1 1: $\neg(\Diamond\Box p \leftrightarrow \Diamond\Box p)$

||

2(*PB*)	1: $\Diamond\Box p$	1: $\neg\Diamond\Box p$	3(*PB*)
4($\leftrightarrow$,1,2)	1: $\neg\Diamond\Box p$	1: $\Diamond\Box p$	5($\leftrightarrow$,1,3)
(2,4)	$\otimes$	$\otimes$	(3,4)

(Assuming they're not going to insist on closing on literals!! -- this is just p <-> p!)

Models:
for (ii), W = {1,2,3}, R ={ (1,2), (2,3)}, ||p|| = {1,3}
for (iii), W = {1,2,3,4}, R ={ (1,2), (2,3), (2,4) }, ||p|| = {1,2,4}

ANSWER 5

MARKING SCHEME
(a) 5 marks
(b) 5 marks
(c) 5 marks
(d) 5 marks

(a)

reactivity	perceives its environment and responds in a timely fashion to changes in the environment
autonomy	operates without being contingent on human direction, is responsible for (most of) its own decision making, has some control over its own actions and internal state
proactivity	exhibits goal-directed behaviour (may have some representation of user's goals), can take initiatives
continuity	is a continuously running process rather than a one-off computation
interactivity	communicates with other agents in some well-defined language, communicates with user(s) through appropriate interface
adaptivity	changes behaviour based on previous experience, new or altered requirements, changes in the environment
mobility	ability to transport itself from one machine to another across a network
rationality	will act so as to achieve its goals, won't act so as not to achieve them
character	believable "personality" and emotional state
orientation	benevolence (do what is required, won't stop others doing what required (without good cause)), veracity (always tells truth)…
reflectivity	exhibits self-awareness by introspection of its own internal state which is used to inform action

(b)
personal service agent attributes:

continuity: needs to be constantly operating to act on behalf of its user

reactivity: needs to be aware of new or changed services, needs to respond to transient system conditions

proactivity: needs to take actions based on understanding of user's goals, may refine search conditions to match other relevant information

adaptivity: needs to learn new behaviour based on user's requirements

interactivity: might need to interact with other user's personal service agents, needs to interact with databases, web sites , etc, but maybe also other search agents

orientation: veracity – it needs to communicate "truth" to the user

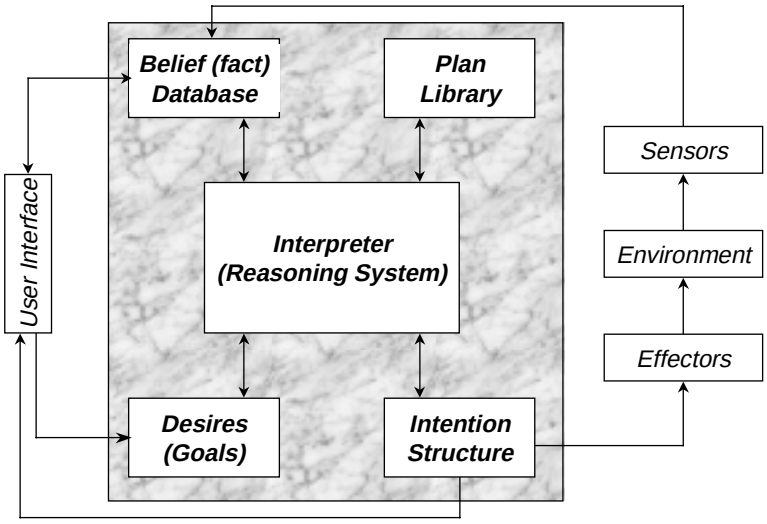
autonomy: needs to undertake search without user direction

mobility: may need to be able to 'move around' network between databases
rationality: will act to satisfy search conditions; won't act so as not to.

(c)

potentially many different ways for environment to change
potentially many different actions system components can take
potentially many different objectives to achieve
actions that best achieve objectives depend on state of environment
environment only sensed locally
speed of computation bounded by rate of change of environment
Examples: air traffic control, navigating underground, ...

(d)



Main Components

- belief database: facts about the 'world'
- desires: goals to be realized
- plan library: sequences of actions to achieve goals
- intentions: those plans chosen for execution
- interpreter: executes intentions, updates beliefs, modifies goals chooses plans

Interpreter Execution Cycle

- At time t: certain beliefs are held, goals are established, plan (partially) executed
- Events occur: either sensed in environment, or via communication, which alter beliefs or modify goals
- Combination of goals and beliefs will trigger action plan(s)
- One or more will be chosen and put on intention structure
- Interpreter selects an executable plan and executes one step
- Cycle repeats

ANSWER 6

MARKING SCHEME
(a) 5 marks
(b) 5 marks
(c) 6 marks
(d) 4 marks

(a)
KQML Brief Description
- Formal definition of message format and message handling protocols
- Allows two or more intelligent systems to share knowledge and pool resources to support cooperative problem solving

Conceptual Layers
- mechanics of communication, via a set of features encoding lower-level communication parameters;
- logic of communication (message layer), via an extensible set of message primitives called performatives
- content of communication, opaque to the language, can be of any form, type indicated by an attribute in the message layer

Syntactic Form and Semantic Model
Kqml_Msg ::= performative Attribute_Value_List
Attribute_Value_List ::= ':' attribute value
| ':' attribute value Attribute_Value_List
Semantic model: agents have belief states and goal stores, messages affect these in certain ways (e.g. tell affects beliefs, request affects goals, etc.)

(b)
Facilitator role to provide meta communication services like well known ports in TCP/IP networks, so supporting directory services and other third party protocols like subscribe, recommend, recruit and broker

Assume Service S is to inform about the status of some proposition/property/commodity X.

subscribe			recommend		
A	-- subscribe(S) -->	F	B	-- advertise(S) -->	F
B	-- tell(X) -->	F	A	-- recommend(S) -->	F
F	-- tell(X) -->	A	F	-- reply(B) -->	A
			A	-- ask(X) -->	B
			B	-- tell(X) -->	A

No. Doesn't say what happens if A recommends before advertise, or B withdraws advertise. In subscribe, B is telling information it might believe, hence in (c)...

(c)
Let knowledge K be a shorthand for: $K_s p \leftrightarrow B_s p \vee B_s \neg p$

$\models B_s p \wedge D_s B_r p \rightarrow I_s\langle inform(r,p)\rangle$

s believes p and wants (desires) r to believe p, then s intends to inform r of p

$\models D_s K_s p \wedge B_s K_r p \rightarrow I_s\langle query(r,p)\rangle$
s wants to know p and believes that r knows p, then s intends to query r about p

$\models D_s DONE(A) \wedge B_s Capability(r,A) \rightarrow I_s\langle command(r,A)\rangle$
s wants action A done and believes r is capable of A, then s intends to command r to do A

$[s,inform(r,p)]\ B_r p$
after s informs r that p, r believes p

$[s,query(r,p)]\ (B_r p \rightarrow I_r\langle inform(s,\underline{true})\rangle) \otimes (B_r \neg p \rightarrow I_r\langle inform(s,\underline{false})\rangle)$
after s queries r about p, then either r believes p and intends to inform s that it is true, or r does not believe p and intends to inform s that it is false

$[s,command(r,A)]\ I_r\langle A\rangle \wedge (DONE(A) \rightarrow I_r\langle inform(s, DONE(A))\rangle)$
after s commands r to do A, then r should intend to do A, and after DONE(A) become true, r should intend to informs that DONE(A) is true

(c)
sincerity: for informs, sincerity is built into the axioms: agents believe what they say
other axiomatisations are possible, e.g.:
$\models D_s P \wedge B_s(DONE(A) \rightarrow P) \rightarrow I_s\langle A\rangle$
i.e. if s wants P and believes that doing A will achieve P, then s will intend to do A. So if P is Brp, then A will be inform(r,p)

trust: put an extra condition on the action modality for believing informs, e.g.:
$[s,inform(r,p)]\ trust(r,s) \rightarrow B_r p$
after s informs r that p, if r trusts s then r believes p
alternatively we could label formulas with their 'trustworthiness'

authority: could be some requirement to negotiate authority relationships for the suspension of autonomy, so one might establish a permission to command r to to A, otherwise it would have to request it (and hope for the best):
$\models D_s DONE(A) \wedge B_s Capability(r,A) \wedge P(auth(s,r)) \rightarrow I_s\langle command(r,A)\rangle$
or
$\models D_s DONE(A) \wedge B_s Capability(r,A) \rightarrow I_s\langle request(r,A)\rangle$
The action modality might be effected to become:
$[s,command(r,A)](P(auth(s,r)) \rightarrow I_r\langle A\rangle) \wedge (DONE(A) \rightarrow I_r\langle inform(s, DONE(A))\rangle)$

belief revision: There is an asymmetry for the results of an inform for a sending agent and a beleiving agent. The receiving agent may believe the content p according to trust, etc., the sending agent can assume that the receiving agent may believe it, but may have to revise this belief if it turns out that the receiver did not trust the sender.

THIS PAGE IS OTHERWISE BLANK AND SIGNIFIES END OF THE EXAM SCRIPT AND MODEL ANSWERS FOR
E3.16 ARTIFICIAL INTELLIGENCE.

[END]