UNIVERSITY OF EDINBURGH
COLLEGE OF SCIENCE AND ENGINEERING
SCHOOL OF INFORMATICS

Date: 24 August 2005 **Time: 09:30-11:30**

ARTIFICIAL INTELLIGENCE 2Bh

Examiners	J. Barnden	(External)
	H. Thompson	(Chair)

INSTRUCTIONS TO CANDIDATES

1. Candidates in the third or later years for the degree of M.A. (Arts), B. Comm, B.Sc (Social Sciences) or LL.B should put a cross (X) in the box on the front cover of the script book.
2. **Answer Parts A and B.** Parts A and B are worth 50% each. Answer ONE question from Part A and ONE question from Part B. If you attempt additional questions in either Part A or Part B, cross out one answer; if you do not, then the examiners will cross out the last question you answered in each Part.
3. Answer Parts A and B in separate script books – one script book for each Part.

Write as legibly as possible.

CALCULATORS ARE NOT PERMITTED.

THIS EXAMINATION WILL BE MARKED ANONYMOUSLY

Part A
ANSWER ONE QUESTION FROM PART A

1. (a) Provide brief (2-3 sentences) explanations for the following terms within the context of learning from data:

   (i). Linearly separable

   (ii). Ockham's razor [6%]

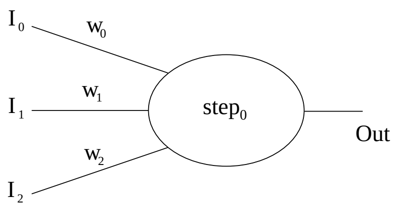
   (b) During the construction of a decision tree, a split on an attribute A with 3 values yields 3 subsets such that

   - subset 1 has 4 positive examples only
   - subset 2 has 2 negative examples and 6 positive examples
   - subset 3 has 2 negative examples and 2 positive examples

   (i). Give a formula for the average entropy *after* the split on A. [5%]

   (ii). Give an expression for the *information gain* after the split on A in terms of the entropy formula $I(p, n)$ for a set S, where p is the number of positive examples and n the number of negative examples in S. You do *not* need to compute the value of the expression. [5%]

   (c) Consider the following simple perceptron:



   in which

   - I_1 and I_2 are the inputs, and Out is the output.
   - $step_0$ is a step function (threshold 0) which acts as its activation function:

$$step_0(z) = \begin{cases} 0 & \text{if } z < 0 \\ 1 & \text{if } z \geq 0 \end{cases}$$

   - I_0 is the extra input (bias) whose value is always -1.

   Indicate what changes are made to the weights w_0, w_1, and w_2 by the perceptron learning algorithm in response to each example A, B, C in the following sequence of training data. You should assume that w_0 is initially 1, w_1 is initially 0, w_2 is initially 0, and that the learning rate is 1.

Example	I_0	I_1	I_2	T
A	-1	0	0	0
B	-1	0	1	1
C	-1	1	0	0

   Note: T denotes the desired (target) output. [9%]

(d) Recall the backpropagation learning algorithm for multi-layer neural networks. You may assume that

- each labelled example is a pair (I,T), where I is the input vector and T is the *desired* output
- O denotes the output for each output unit
- g denotes the differentiable everywhere activation function
- W_{ji} denotes the weight on link from unit j to unit i.

(i). How is the neural network initialised?

[3%]

(ii). Give the formulae for the observed error and Δ_i for each output unit i.

[8%]

(iii). Briefly describe the backpropagation step of the learning algorithm. Your answer should include the formula for the error term at each hidden layer.

[8%]

(iv). Give the update rule for the weights of the network. You should explain any new terms that you introduce.

[6%]

2. (a) Provide brief (2-3 sentences) explanations for the following terms.

(i). Decision networks

(ii). Utility node in a decision network

[8%]

(b) Outline the algorithm used for evaluating a decision network.

[8%]

(c) (i). Define conditional probability.

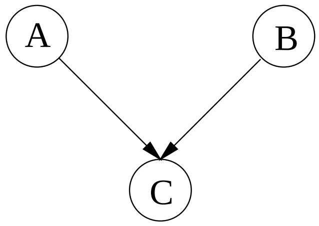
(ii). State Bayes' theorem.

(iii). Prove that

$$\texttt{Pr}\{X, Y|Z\} = \texttt{Pr}\{X|Y, Z\}\texttt{Pr}\{Y|Z\}$$

[12%]

(d) Consider the following Bayesian network with binary variables A, B, and C, and associated conditional probability tables:



$\texttt{Pr}\{A = 0\}$	$\texttt{Pr}\{A = 1\}$
1/2	1/2

$\texttt{Pr}\{B = 0\}$	$\texttt{Pr}\{B = 1\}$
2/3	1/3

A	B	$\texttt{Pr}\{C = 0\}$	$\texttt{Pr}\{C = 1\}$
0	0	3/4	1/4
0	1	1/2	1/2
1	0	1/2	1/2
1	1	0	1

Derive the value of the following probability:

- $\texttt{Pr}\{C = 0|A = 0\}$

You should explain your computations, writing appropriate formulae where needed, and only then substituting numbers from the tables.

[9%]

(e) Consider the following situation involving stochastic simulation by likelihood weighting:

During sampling, a node X, corresponding to an evidence variable with given value x, is chosen. If $Y = y$ and $Z = z$ are values chosen for parent nodes of X, state the weight w associated with the current simulation round.

[4%]

(f) In the context of utility theory, assume that $Result(Act)$ denotes the set of possible outcomes of an action Act and $U(X)$ denotes the utility of X:

(i). What is the expected utility of Act given evidence E?

[3%]

(ii). What is the expected value of the current best action α_v, given current knowledge E and the fact that a *sense operation* on one of the variables X_i yielded the value v?

[3%]

(iii). Briefly explain what is meant by the *value of perfect information.*

[3%]

Part B
ANSWER ONE QUESTION FROM PART B

1. (a) Let $\boxed{\mathsf{K}_A}\,\varphi$ be a modal operator for knowledge, meaning that agent A knows φ. Let w_0 be a possible world, $\equiv_A$ be the accessibility relation for A between possible worlds and let $w_0 \vDash \varphi$ mean that φ is true in world w_0. Give a definition of $w_0 \vDash \boxed{\mathsf{K}_A}\,\varphi$.

[3%]

(b) Try to formalise the following quotation of Donald Rumsfeld as a modal logic formula. Represent "we know φ" as $\boxed{\mathsf{K}_{we}}\,\varphi$.

As we know,
There are known knowns.
There are things we know we know.
We also know
There are known unknowns.
That is to say
We know there are some things
We do not know.
But there are also unknown unknowns.
The ones we don't know we don't know.
(*The Guardian*, Saturday May 3, 2003, p. 13)

[Hint: You will need to quantify over propositions, e.g. $\exists\varphi$]

[6%]

(c) Property T asserts that anything an agent knows is true. Formally,

$$\boxed{\mathsf{K}_A}\,\varphi \implies \varphi$$

What attribute must be true of the accessibility relation for A for Property T to hold? Prove that Property T holds when this attribute is true.

[8%]

(d) How realistic is Property T as part of a model of human knowledge?

[5%]

(e) Define the modal operators of "everybody knows that" (written $\boxed{\mathsf{E}}_s\varphi$) and "it is common knowledge that" (written $\boxed{\mathsf{C}}_s\varphi$) with respect to a set of agents $s = \{A, B, C, \ldots\}$.

[6%]

(f) In the Coordinated Attack problem two generals try to reach agreement to attack a castle from different directions at the same time. Common knowledge of the agreed plan is essential if both generals are to be confident that the attack will be simultaneous. Assume that communication consists of a k-round handshake protocol, i.e. General A sends a message to General B, then B to A, then A to B, etc, with each message taking a finite time. Suppose that the first message is "Attack at dawn", the second message is "I agree" and subsequent messages are "I got your previous message". There is no guarantee that every message will be received. Explain, using the modal logic of knowledge, why common knowledge is unattainable under these conditions.

[6%]

(g) Consider the communicative action $Query_{s,h}\varphi$ in which s asks h whether φ is true and h is expected to give a yes/no reply. Represent this action with a STRIPS-like operator. Explain the role of the *happy* predicate and explain why the STRIPS add and delete lists have to be merged into a single effects list.

[10%]

(h) Equipping an agent with communicative actions, such as *Query* and *Inform*, in addition to regular actions, such as *go*, *pickup*, etc, facilitates the formation of multi-agents plans. Give two examples of practical applications of such multi-agent planning and discuss the role of the communicative actions within these plans.

[6%]

2. A simple Lift World can be defined using the following functions and predicates.

- *Elli* is the lift agent and *Lift* is the location inside the lift.
- $Loc(a, p, s)$ means that agent a is located at place p in situation s.
- $Door(status, s)$ means the lift door is open if $status = O$ and closed if $status = C$ in situation s.
- $Go(f_1, f_2)$ is the action of the lift going from floor f_1 to floor f_2.
- $Enter(a)$ is the action of agent a entering the lift.
- $Exit(a)$ is the action of agent a exiting the lift.
- *Open* is the action of opening the lift door.
- *Close* is the action of closing the lift door.
- $Result(act, s)$ is the situation resulting from executing the action act in situation s.

(a) Give Situation Calculus EFFECT rules for the actions $Enter(a)$ and $Go(f_1, f_2)$.
[10%]

(b) What is a FRAME RULE and why are they needed in Situation Calculus? Illustrate your answer with frame rules for the actions *Open* and $Exit(a)$.
[10%]

(c) Give STRIPS operator tables for the actions $Go(f_1, f_2)$, $Exit(a)$ and *Close*. Explain any necessary changes to the functions and predicates as we move from the Situation Calculus representation to the STRIPS one. Explain how STRIPS avoids the need for frame rules.
[12%]

(d) Give a STRIPS plan to describe how *Alan* can go from inside the lift to floor 3, where the initial state is that *Alan* is in the lift with the door closed and *Elli* is at floor 3. Describe the initial situation using a knowledge base. Describe the process by which STRIPS might form a plan to achieve the goal $Loc(Alan, 3)$ using a situation-based search starting from the initial situation. You can assume that STRIPS always make the correct guess at any branching point in the search.
[10%]

(e) Consider the more difficult problem of trying to solve the goal $Loc(Alan, 3)$ from the initial situation where *Alan* and *Elli* are at floor 1 and the door is open. The planning process will initially proceed as before, but a new subogoal arising from the preconditions of $Exit(Alan)$ is $Door(O)$. We actually want this subgoal to be achieved by the action *Open*, but it is already true in the initial situation, so it will appear that nothing needs to be done. Partial-order planning provides machinery to solve such difficulties. Explain this machinery and how it would work in this problem.
[8%]

Specimen Answers

Part A

1. (a) **Linearly separable:**
The n inputs to a neural network can be thought of as points in a n dimensional plot. If the unit has to produce 0-1 output, then the problem is said to be linearly separable if the points for which 0 is required can be separated from the other points by a hyperplane. Perceptrons can only solve linearly separable problems.
Ockham's razor:
Inductive learning involves finding a consistent hypothesis that agrees with the (training) examples. Ockham's razor suggests choosing the simplest consistent hypothesis (e.g. the most concise decision tree).
Marking guide: 3 marks each

(b) (i). The average entropy *after* the split on A is given by

$$Remainder(A) = \sum_{i=1}^{3} \frac{p_i + n_i}{p + n} I(p_i, n_i)$$

where p_i and n_i are the number of positive and negative examples, respectively, in subset i after the split, and p and n are the number of positive and negative examples, respectively, before the split.
(ii).
The information gain (improvement) after the split is given by

$$Gain(A) = I(p, n) - Remainder(A)$$

Thus, for given example, we have
Gain(A) $= I(12, 4) - (\frac{4}{16} I(4, 0) + \frac{8}{16} I(6, 2) + \frac{4}{16} I(2, 2)) = I(12, 4) - (0 + \frac{1}{2} I(6, 2) + \frac{1}{4}(1)) = I(12, 4) - (\frac{1}{2} I(6, 2) + \frac{1}{4})$
Marking guide: Give marks as indicated for each part of question.

(c) In what follows:
The update, given in vector notation, is as follows (where η is the learning rate):

$$\mathbf{W} \leftarrow \mathbf{W} + \eta \mathbf{I}(T - Out)$$

Example A:
$\sum_{j=0}^{2} w_j I_j = (1)(-1) + (0)(0) + (0)(0) = -1 < 0$, so actual output is 0 ($= T$) and the weights are not updated.
Example B:
$\sum_{j=0}^{2} w_j I_j = (1)(-1) + (0)(0) + (0)(1) = -1 < 0$, so actual output is 0 (while $T = 1$). Therefore, weights are updated:

$$(w_0, w_1, w_2) \leftarrow (1, 0, 0) + (1)(-1, 0, 1)(1 - 0) = (0, 0, 1)$$

Example C:
$\sum_{j=0}^{2} w_j I_j = (0)(-1) + (0)(1) + (1)(0) = 0 \geq 0$, so actual output is 1 (while $T = 0$). So, weights are updated.

$$(w_0, w_1, w_2) \leftarrow (0, 0, 1) + (1)(-1, 1, 0)(0 - 1) = (1, -1, 1)$$

Marking guide: 3 marks per example.

(d) The update rules for backpropagation on slide 4-45 and the pseudo-code given on slide 4-46 provide the answers for the various parts.

(i). *Initialization:* The algorithm starts by assigning arbitrary weights to the network.

(ii). The basic operation of the algorithm deals with a single example. On example (I,T), the algorithm first computes the output O for each output unit i. It then computes the error $Err = T - O$ and $\Delta_i = g'(in_i) \cdot Err$, where in_i is the total weighted input to output unit i.

(iii). *Backpropagation Step:* The algorithm (now) iterates over each previous layer in network and computes the Δ values for each hidden unit i in the layer. The computation is given by $\Delta_i = g'(in_i) \sum_k W_{ik} \Delta_k$, where k ranges over all the units connected to i's output.

(iv). *Update weights:* Once backpropagation has reached input layer, update all weights using gradient descent $W_{ji} \leftarrow W_{ji} + \eta a_j \Delta_i$, where η is the learning rate.

Marking guide: Give marks as indicated for each part of question.

2. (a) **decision networks:**

These provides a simple formalism for expressing and solving decision problems. They are a natural extension of Bayesian networks, containing decision, utility, and chance nodes.

utility node:

A utility node represents the agent's utility function. The parents of the node are the variables describing the outcomes that directly affect utility. Associated with the utility node is the agent's utility given as a function of the parent attributes.

Marking guide: 4 marks each

(b) An adequate answer will be along those lines:

For evaluating what action to take:

1. Set the evidence variables for the current state
2. For each possible value of decision node:
   a). Compute posterior probabilities $\mathbf{Pr}(\texttt{results}|\texttt{evidence and decision node})$
   b). Compute the expected utility of the action given the evidence
3. Return action with highest expected utility

Marking guide: 2 marks for step 1; 4 marks for step 2 and 2 marks for step 3

(c) Conditional Probability:

If A and B are two events and $\texttt{Pr}\{B\} \neq 0$, then the probability of A given B has already occurred is written $\texttt{Pr}\{A|B\}$ and defined as

$$\texttt{Pr}\{A|B\} = \frac{\texttt{Pr}\{A \cap B\}}{\texttt{Pr}\{B\}}$$

Bayes' Theorem:

$$\texttt{Pr}\{A|B\} = \frac{\texttt{Pr}\{A\}\texttt{Pr}\{B|A\}}{\texttt{Pr}\{B\}}$$

Proof that $\texttt{Pr}\{X,Y|Z\} = \texttt{Pr}\{X|Y,Z\}\texttt{Pr}\{Y|Z\}$:

$$\texttt{Pr}\{X|Y,Z\}\texttt{Pr}\{Y|Z\} = \frac{\texttt{Pr}\{X,Y,Z\}}{\texttt{Pr}\{Y,Z\}} \cdot \frac{\texttt{Pr}\{Y,Z\}}{\texttt{Pr}\{Z\}} = \frac{\texttt{Pr}\{X,Y,Z\}}{\texttt{Pr}\{Z\}}$$

$$= \frac{\texttt{Pr}\{X,Y|Z\}\texttt{Pr}\{Z\}}{\texttt{Pr}\{Z\}} = \texttt{Pr}\{X,Y|Z\}$$

Marking guide: Give 4 marks for each.

(d) Computing requested probability:

$$\begin{aligned}
\texttt{Pr}\{C=0|A=0\} &= \sum_v \texttt{Pr}\{C=0, B=v|A=0\} \text{ (by def. of marginal distribution)}\\
&= \frac{\sum_v \texttt{Pr}\{C=0,B=v,A=0\}}{\texttt{Pr}\{A=0\}} \text{ (by conditional probability rule)}\\
&= \frac{\sum_v \texttt{Pr}\{C=0|B=v,A=0\}\texttt{Pr}\{B=v,A=0\}}{\texttt{Pr}\{A=0\}} \text{ (cond. prob.)}\\
&= \frac{\sum_v \texttt{Pr}\{C=0|B=v,A=0\}\texttt{Pr}\{B=v|A=0\}\texttt{Pr}\{A=0\}}{\texttt{Pr}\{A=0\}} \text{ (cond. prob.)}\\
&= \sum_v \texttt{Pr}\{C=0|B=v,A=0\}\texttt{Pr}\{B=v|A=0\}\\
&= \sum_v \texttt{Pr}\{C=0|B=v,A=0\}\texttt{Pr}\{B=v\} \text{ (as } B \text{ is independent of } A)\\
&= \texttt{Pr}\{C=0|B=0,A=0\}\texttt{Pr}\{B=0\}\\
&\quad +\texttt{Pr}\{C=0|B=1,A=0\}\texttt{Pr}\{B=1\}\\
&= (1/4)(1/3) + (1/2)(2/3) = 5/12
\end{aligned}$$

Marking guide: Marks as indicated

(e) $w \leftarrow w \cdot \texttt{Pr}\{X=x|Y=y,Z=z\}$

Marking guide: Give marks as indicated.

(f) (i). $EU(Act|E) = \sum_{R\in Result(Act)} \texttt{Pr}\{R|E,Act\}U(R)$

(ii). $EU(\alpha_v|E,X_i=v) = \sum_{R\in Result(Act)} \texttt{Pr}\{R|E,Act,Xi=v\}U(R)$

(iii). The value of perfect information gives the expected improvement in utility after finding some information compared with making a decision without the information.

Marking guide: Give marks as indicated for each part of question.

Part B

1. (a) $w_0 \models \boxed{\mathsf{K}_A}\,\varphi \iff \forall w.w_0 \equiv_A w \implies w \models \varphi$

Marking guide: Lose a mark or half mark for each omission or unwanted inclusion. Lose a mark for each significant error in the definition.

(b) There are several acceptable variants. This is one.

$$\begin{aligned}
&\boxed{\mathsf{K}_{we}}\,(\exists\varphi\,\boxed{\mathsf{K}_{we}}\,\boxed{\mathsf{K}_{we}}\,\varphi)\wedge\\
&\quad(\exists\varphi\,\boxed{\mathsf{K}_{we}}\,\neg\boxed{\mathsf{K}_{we}}\,\varphi)\wedge\\
&\quad(\exists\varphi\,\neg\boxed{\mathsf{K}_{we}}\,\neg\boxed{\mathsf{K}_{we}}\,\varphi)
\end{aligned}$$

Marking guide: 2 marks for each conjunct. Lose a mark or half mark for each omission or unwanted inclusion.

(c) The accessibility relation must be reflexive.

Suppose:	$w_0 \models \boxed{\mathsf{K}_A}\,\varphi$
by meaning $\boxed{\mathsf{K}_A}$:	$\forall w.\ w_0 \equiv_A w \implies w \models \varphi$
since $\equiv_A$ is reflexive:	$w_0 \models \varphi$
discharging assumption:	$\boxed{\mathsf{K}_A}\,\varphi \implies \varphi$

Marking guide: 2 marks for knowing the attribute. Lose a mark for failing to mention reflexivity or for including additional, unnecessary properties. 6 marks for the proof.

(d) Property T assumes that an agent has infallible knowledge of the world, but we know that humans are fallible. If we restrict knowledge to those beliefs that happen to be true, then Property T is true by definition, but this shifts the problem to one of detecting which of an agents's beliefs are true and, therefore, qualify as knowledge.

Marking guide: This is an open-ended question and more than one conclusion is possible. Marks will be awarded for the coherence of the arguments supporting whatever position is taken.

(e)

$$\boxed{\mathsf{E}}_s\varphi \;\Leftrightarrow\; \boxed{\mathsf{K}_A}\,\varphi \wedge \boxed{\mathsf{K}_B}\,\varphi \wedge \boxed{\mathsf{K}_C}\,\varphi \ldots$$
$$\boxed{\mathsf{C}}_s\varphi \;\Leftrightarrow\; \boxed{\mathsf{E}}_s\varphi \wedge \boxed{\mathsf{E}}_s\boxed{\mathsf{E}}_s\varphi \wedge \boxed{\mathsf{E}}_s\boxed{\mathsf{E}}_s\boxed{\mathsf{E}}_s\varphi \ldots$$

Marking guide: 3 marks for $\boxed{\mathsf{E}}$ *and 3 marks for* $\boxed{\mathsf{C}}$. *Lose a mark for each omission or unwanted inclusion. Lose a mark for each significant error in representation.*

(f) After the first message $\boxed{\mathsf{K}_B}\,m$ holds, after the second message $\boxed{\mathsf{K}_A}\boxed{\mathsf{K}_B}\,m$ holds, and after the $2n^{th}$ message $(\boxed{\mathsf{K}_A}\boxed{\mathsf{K}_B})^n\ m$ holds. However, common knowledge requires $(\boxed{\mathsf{K}_A}\boxed{\mathsf{K}_B})^n\ m$ to hold for all n and this situation can never be reached in a finite time.

Marking guide: 4 marks for the essential idea of an interleaving of modal operators and 2 marks for using this to conclude that common knowledge is unattainable.

(g) The yes/no *Query* action can be described by the following STRIPS-like operator table.

act:	$\mathtt{Query}_{s,h}(\varphi)$
pre:	$\boxed{\mathsf{K}_s}\,\left(\boxed{\mathsf{K}_h}\,\varphi \vee \boxed{\mathsf{K}_h}\,\neg\,\varphi\right)$
eff:	$\boxed{\mathsf{K}_h}\,\left(\left(\boxed{\mathsf{K}_s}\,\varphi \vee \boxed{\mathsf{K}_s}\,\neg\,\varphi\right) \rightarrow \text{happy}(s)\right)$

The *happy* predicate is a trick which allows agent s to dynamically assign a new goal to agent h. Agent h is given a single static goal to make s happy. The effect of the *Query* action is to make h know that if s knows either φ or $\neg\,\varphi$ then this knowledge will make s happy. h will then infer a subgoal of ensuring that s

knows this information. To describe this effect we need a compound formula that cannot be divided into a simple conjunction of facts to be added or deleted. So the STRIPS operator representation must be generalised to allow such compound effects.

Marking guide: 3 marks each for the effect and precondition lists. Lose a mark for each omission or unwanted inclusion. Lose a mark for each significant error in representation. 2 marks for explaining the role of happy and 2 marks for the explanation of the merging of add and delete lists into an effects formula.

(h) This is an open-ended question. Examples might include: negotations between softbots on the web, e.g. buying and selling goods; teams of robots manipulating physical objects; adversarial agents, e.g playing a game such as chess or poker; etc.

Marking guide: 3 marks for each example. 2 marks for the description of each example and 1 mark for the description of the role of communication in each example.

2. (a)

$$\begin{aligned}\forall a, f, s.\ Loc(a, f, s) \wedge Loc(Elli, f, s) \wedge Door(O, s) \quad &\Longrightarrow \\ Loc(a, Lift, Result(Enter(a), s)) \wedge \neg Loc(a, f, Result(Enter(a), s))& \\ \forall f_1, f_2, s.\ Loc(Elli, f_1, s) \wedge Door(C, s) \quad &\Longrightarrow \\ Loc(Elli, f_2, Result(Go(f_1, f_2), s)) \wedge \neg Loc(Elli, f_1, Result(Go(f_1, f_2), s))&\end{aligned}$$

Marking guide: 5 marks for each rule. Lose a mark or half mark for each omission or unwanted inclusion. Lose a mark for each significant error in the definition.

(b) Frame rules describe what facts are *unaffected* by an action. For instance,

$$\begin{aligned}\forall a, f, s.\ Loc(a, f, s) \quad &\Longrightarrow \quad Loc(a, f, Result(Open, s)) \\ \forall a, a', f', s\ Loc(a', f', s) \wedge a' \neq a \quad &\Longrightarrow \quad Loc(a', f', Result(Exit(a), s)) \\ \forall a, s\ Door(Open, s) \quad &\Longrightarrow \quad Door(Open, Result(Exit(a), s))\end{aligned}$$

They are needed in Situation Calculus because the planner needs to infer both what has changed and what is unchanged by an action in order to have a full model of the situation resulting from the action. For instance, if the planner's goal is conjunctive and it achieves the first conjunct then it needs to be able to conclude that the first conjunct is still true when the second one is achieved.

Marking guide: 2 marks for the definition of a frame rule. 3 marks for each of two example frame rules. 2 marks for explaining why they are needed.

(c) The operator tables for $Go(f_1, f_2)$, $Exit(a)$ and $Close$ are as follows:

act:	$Go(f_1, f_2)$
pre:	$Loc(Elli, f_1) \wedge Door(C)$
add:	$Loc(Elli, f_2)$
del:	$Loc(Elli, f_1)$

act:	$Exit(a)$
pre:	$Loc(a, Lift) \wedge Loc(Elli, f) \wedge Door(O)$
add:	$Loc(a, f)$
del:	$Loc(a, Lift)$

act:	*Close*
pre:	$Door(O)$
add:	$Door(C)$
del:	$Door(O)$

STRIPS represents situations with a series of knowledge bases. Each new knowledge base is calculated from the last one by assuming that everything remains the same, except when it is explicitly added from the add list or deleted from the delete list. So explicit frame rules are replaced by an implicit "closed world" assumption that facts remain true in the new situation unless told otherwise.

Marking guide: 3 points for each of the operator tables and 3 points for the explanation of how STRIPS *avoids frame rules.*

(d) **Plan:** $Open, Exit(Alan)$.

Initial State: $Door(C), Loc(Alan, Lift), Loc(Elli, 3)$.

Planning Process:

(i). Attempt to match $Loc(Alan, 3)$ in the initial state. This fails.

(ii). Identify an operator with an add list conjunct matching this goal. Find $Exit(Alan)$. The preconditions of this action, $Loc(Alan, Lift) \wedge Loc(Elli, 3) \wedge Door(O)$, become the new subgoals.

(iii). $Loc(Alan, Lift)$ and $Loc(Elli, 3)$ both match with facts in the initial situation, leaving only the goal $Door(O)$, which is not true in the initial situation.

(iv). Seek an operator with an add list conjunct which matches $Door(O)$. Find *Open*. The preconditions of this action, $Door(C)$ becomes a new subgoal.

(v). $Door(C)$ is true in the initial situation.

(vi). All the preconditions of *Open* are now true in the initial situation, so it can be executed to form the new situation: $Door(O), Loc(Alan, Lift), Loc(Elli, 3)$.

(vii). All the preconditions of $Exit(Alan)$ are now true in the new situation, so it can be executed to form the final situation: $Door(O), Loc(Alan, 3), Loc(Elli, 3)$. The original goal is now achieved, so the process outputs the above plan and halts.

Marking guide: 2 marks for the plan; 2 marks for the initial situation and 6 marks for the description of the planning process.

(e) Partial order planning proceeds by the incremental development of plans via search in plan space. It provides machinery for protecting facts that are needed as the preconditions of later actions. It detects conflicts between actions, for instance, where one action will delete a protected fact that is needed as the precondition for another action. It tries to resolve these conflicts by reordering the clashing actions — either demoting the fact-deleting action to before the achievement of the protected fact or promoting it until after the second action has been executed and the protected fact is no longer needed. In this case the protected fact is $Door(Open)$, which is a precondition of $Exit(Alan)$. The fact-deleting action is *Close*, which will be required to achieve the subgoal $Door(C)$, arising as a precondition of $Go(1, 3)$. Since $Exit(Alan)$ has to be the last action of the plan, promotion of *Close* will not work. So demotion is required. But this, in turn, requires an additional *Open* action to be inserted to reinstate $Door(O)$ after it has been deleted by *Close*.

Marking guide: This is a deliberately challenging question, intended to distinguish the high-flyers. It requires a deep understanding of how partial-ordered planning differs from situation-based planning and how conflicts arise and are resolved. In addition, the example problem is quite complicated, so a lot needs to be kept in mind. Consequently, the marking will be generous if anyone shows some understanding of the issues. 4 marks for an explanation of the conflict-resolution machinery in partial-order planning and 4 marks for the explanation of how it addresses this case.