C1 Artificial Intelligence (25 points) (Do all parts.)
Search

You are given a sequence of letters and blanks (such as the sequence `''ahrll arlls in our fbrlsa''`) and must decrypt it to produce a sequence of valid words. An operator consists of selecting a letter in the current sequence and replacing every occurrence of it with a new letter; the cost of the operator is the number of occurrences of the letter. So a solution to the problem of transforming `''ahrll arlls in our fbrlsa''` into a sequence of valid words is:

Substitute `t` for `a` : cost 3
Substitute `o` for `b` : cost 1
Substitute `e` for `l` : cost 5

Producing the sequence `''three trees in our forest''`.

(a) [2 points] What is meant by an admissible heuristic?

(b) [5 points] Consider the following heuristic:
h_1 = number of strings of letters that are not yet valid words
Thus, h_1(`ahrll arlls in our fbrlsa`)$= 3$
Prove either that h_1 is an admissible heuristic or give a counterexample showing that it is not.

(c) [9 points] Prove that the set of states expanded by algorithm A^* for any problem is a subset of the states examined by uniform cost search.

(d) [6 points] Explain how iterative deepening A^* search (IDA^*) works and illustrate it with a sample search space.

(a) [3 points] What advantage does IDA^* have over A^* search? Explain.

C2 Artificial Intelligence (25 points) (Do all parts.)

Game Trees

Consider the following variant of chess: On a player's turn, he does not make a single move, but rather he suggests two possible alternative moves and his opponent may choose between those two. If there is only one legal move, then that move is made.

(a) [15 points] How would the process of game tree evaluation be modified for this new game?

(b) [10 points] There is an analogue of alpha-beta pruning that can be applied to this new game. Give an example of a game tree where pruning is possible. You do not have to give the general rule.

C3 Artificial Intelligence (25 points) (Do all parts.)
Decision Trees

(a) [5 points]
Describe a method for choosing the best attribute to use as the basis of a split in a decision tree.

(b) [13 points]
Consider JR, a janitorial robot that roams the top 3 floors of Smith Hall collecting trash and looking for recyclable materials. The robot is trying to learn which offices have recycling bins and is given the training examples in the table below.

#	STATUS	FLOOR	DEPARTMENT	RECYCLING BIN?
307	faculty	three	POSC	no
309	staff	three	POSC	no
308	faculty	three	CIS	yes
415	student	four	POSC	yes
209	staff	two	CIS	no
217	faculty	two	CIS	yes
316	student	three	POSC	yes
420	staff	four	CIS	no

Notice the attributes describing the office include the STATUS of the occupants (faculty, staff, or student), the FLOOR on which the office is located (two, three, or four), and the DEPARTMENT of the occupants (POSC for Political Science, or CIS for Computer and Information Sciences).

Induce a decision tree which JR can use to decide whether a particular office contains a recycling bin. Show all of your work and be particularly clear about how the best attribute is chosen to root the tree (and its sub-trees).

You may find it useful to recall the following information (I) formulas:

$I\left(\frac{1}{2}, \frac{1}{2}\right) = -\frac{1}{2}log_2\frac{1}{2} - \frac{1}{2}log_2\frac{1}{2} = 1$

$I\left(\frac{1}{3}, \frac{2}{3}\right) = -\frac{1}{3}log_2\frac{1}{3} - \frac{2}{3}log_2\frac{2}{3} = .918$

$I(1,0) = I(0,1) = 0$

(c) [7 points]
Some attributes such as TEMPERATURE have a large or infinite set of possible values. Explain why such attributes might cause problems for decision tree learning. How might these problems be overcome by a decision-tree learning algorithm?

C4 Artificial Intelligence (25 points) (Do all parts.)

Planning

Imagine that we have constructed a software agent that can encapsulate a knowledge base. This agent accepts a query that must contain some GIVEN information, and in return it will PROVIDE some related information. For example, we might have a knowledge base (named "KB25") located at Stanford U. that contains famous AI researchers, and the institutions that they work at, with records like:

```
Researcher-Name: "Stewart Russel"
Institution: "UC Berkeley"
```

We might formally describe this KB and its query schema as follows:

```
Given(kb25,researcher-name)
Provides(kb25,institution)
Institution(kb25,"Stanford U")
```

The nice thing about kb25 is that it knows where everyone works (given a name, it will tell us where they work).

Furthermore, imagine that every university also provides a knowledge base that when given a researcher name, will return the newest paper that the researcher wrote in the form of a URL. You might then use the URL to retrieve the actual paper. For example, the UC Berkeley KB, named "KB02" might have a record like this:

```
Researcher-Name: "Stewart Russel"
Paper-URL: "http://http.cs.berkeley.edu/~russell/papers/bat-final.ps.gz"
```

We might formally describe this KB and its query schema as follows:

```
Given(kb02,researcher-name)
Provides(kb02,paper-url)
Institution(kb02,"UC Berkeley")
```

Of course, the KB located at UC Berkely only knows about papers authored by people at UC Berkeley!

CONTINUED ON NEXT PAGE

CONTINUED: **C4 Artificial Intelligence (25 points) (Do all parts.)**

(a) [20 points] We can use a planning system to construct a series of actions to answer a question like "Get me the newest paper by Stewart Russel". Formally describe (via a STRIPS-like notation) a representative set of actions in this domain, assuming that your local KB has the KB and query schema descriptions mentioned above. Hint: you'll need two actions that gather information before the third action that actually retrieves the paper. Hint: You might start with a fact like `Queried-Researcher-Name("Stewart Russel")` and you might want to end up in a state where the following is true
`Have(?paper) AND Author(?paper,"Stewart Russel")`

(b) [5 points] Imagine instead that the KB at each institution, instead of having just the newest paper by each author, has *all* of the papers by each author, providing both the URL and the DATE of each. Thus querying the KB, given an author, provides a whole set of paper records including a URL and DATE for each paper. What sort of extensions to the simple POP algorithm and STRIPS-style operator descriptions are needed to handle this case?