CS322, Fall 1997
Midterm Examination — Solution

This is a solution to one of the syntactic variants of the midterm. The other exam differed in exactly what the knowledge base was, the goal in the searching question, and the word list for the question in arc consistency.

Question 1 [15 marks]

Consider the knowledge base *KB*:

$$a \leftarrow b \wedge d.$$
$$b \leftarrow e \wedge f.$$
$$c \leftarrow h \wedge e.$$
$$d \leftarrow e.$$
$$d \leftarrow b \wedge g.$$
$$e \leftarrow h.$$
$$g \leftarrow c \wedge d.$$
$$h.$$

(a) [5 marks] a isn't a logical consequence of *KB*. Explain what this means and show why a isn't a logical consequence of *KB*.

(b) [5 marks] g is a logical consequence of *KB*. Give a top-down derivation for the query $?g$.

(c) [5 marks] Give Clark's completion of the knowledge base:

$$passed\_each([], St).$$
$$passed\_each([C|R], St) \leftarrow$$
$$\quad passed(St, C) \wedge$$
$$\quad passed\_each(R, St).$$

Solution to Question 1, part (a)

"a isn't a logical consequence of *KB*" means there exists a model of *KB* in which a is false. To show why a isn't a logical consequence of *KB*, we can give such a model. The minimal model will always be one such model. This model has $\{h, e, d, c, g\}$ all true and $\{a, b, f\}$ all false.

Solution to Question 1, part (b)

Here is a top-down derivation for the query $?g$, where we always select the leftmost atom to resolve against:

$yes \leftarrow g$

$yes \leftarrow c \wedge d$

$yes \leftarrow h \wedge e \wedge d$

$yes \leftarrow e \wedge d$

$yes \leftarrow h \wedge d$

$yes \leftarrow d$

$yes \leftarrow e$

$yes \leftarrow h$

$yes \leftarrow$

Solution to Question 1, part (c)

Clark's completion of the knowledge base:

$passed\_each([], St).$
$passed\_each([C|R], St) \leftarrow$
$\quad passed(St, C) \wedge$
$\quad passed\_each(R, St).$

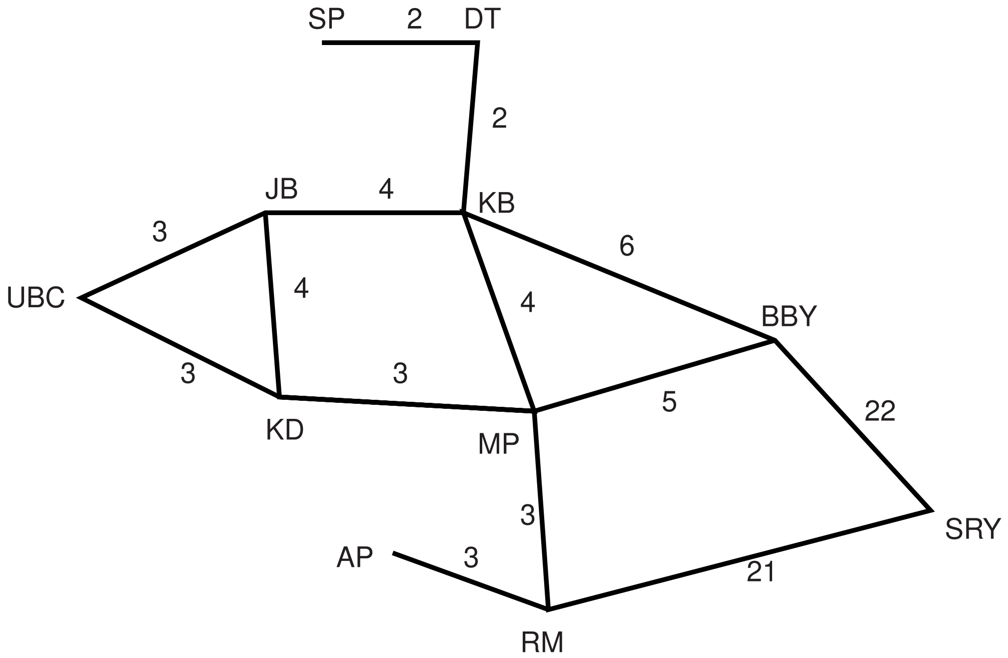
is

$passed\_each(X, Y) \leftrightarrow$
$\quad (\exists St\, X = [] \wedge Y = St) \vee$
$\quad (\exists C \exists R \exists St\, X = [C|R] \wedge Y = St \wedge passed(St, C) \wedge passed\_each(R, St).)$

You could also replace Y everywhere with St, and remove the $\exists St$ and $Y = St$.

Question 2 [15 marks]

Consider the graph (not drawn to scale) with arc lengths shown on the arcs:



Suppose we have the following heuristic values for the distance to *sp*.

h(sp)=0	h(dt)=2	h(kb)=3
h(jb)=3	h(ubc)=5	h(kd)=6
h(mp)=7	h(bby)=8	h(ap)=8
h(rm)=9	h(sry)=29	

(a) [6 marks] Show the nodes expanded (taken off the frontier), in order, and the f-value for each node added to the frontier for an A^* search from *ubc* to *sp*. Assume that multiple-path pruning is used, and that the search stops after the first path is found. Show clearly the path found.
[You can either number the nodes on the graph and give the f-values, or use the space below to write your answer.]

(b) [6 marks] Show how dynamic programming can be used to find a path from *ubc* to *sp*. Show all distance values that are computed, and how these are used to find the shortest path. What path is found?
[You may use the graph to show your answer, or write your answer in the space below, but you must explain clearly what your notation means.]

(c) [3 marks] Suppose you were contacted by TecnoTaxi to advise on a method for finding routes between locations in Vancouver. What method would you recommend, and why. Give one shortcoming of the method you propose. You must use full sentences.

Solution to Question 2, part (a)

The following shows the nodes added to and removed from the queue:

Node	f-value	Order Removed
UBC	5	1
JB	6	2
KD	9	3
KB	10	4
MP	13	
BBY	21	
DT	11	5
SP	11	6

Note that the other tree nodes (AP, RM and SRY) are never places on the queue, and their f-value is not computed.

Solution to Question 2, part (b)

Dynamic programming builds the following distance map to *sp*

Node	Distance
SP	0
DT	2
KB	4
JB	8
MP	8
BBY	10
UBC	11
KD	11
RM	11
AP	14
SRY	32

To get from UBC to SP, you need to compare going via JB (with distance 3+8) with going via KD (distance 3+11). Thus the first step is to JB. From JB, you need to compare going via KB (distance 8), via KD (distance 15) and UBC (distance 14) and choose KB. From KB, you choose DT and then SP. The final route is:

$$UBC \rightarrow JB \rightarrow KB \rightarrow DT \rightarrow SP$$

Solution to Question 2, part (c)

You could have suggested dynamic programming, in that once the distance tables have been computed they can be accessed quickly. The problem is that you need a distance function to every possible destination.

You could have suggested A^* that will find shortest routes. The problems is that the time and space is exponential in the path length.

The best solution is probably some hierarchical route finder, finding distances between major intersections and then locally searching from a particular location to these intersections.

Question 3(a)

Question 3 is worth 15 marks.

(a) Suppose you have a relation $r(X, Y)$ that is true of there is a word in the word list below with first letter X and second letter Y.

The word list is:

add	arc	bad	bud
cup	dip	fad	odd

Suppose the domain of X is $\{a, b, c, d\}$ and that of Y is $\{a, d, i, r\}$.

(i) [3 marks] Is the arc $\langle X, Y\rangle$ arc consistent? If so, explain why. If not, show what element(s) can be removed from a domain to make it arc consistent.

(ii) [3 marks] Is the arc $\langle Y, X\rangle$ arc consistent? If so, explain why. If not, show what element(s) can be removed from a domain to make it arc consistent.

Solution to question 3(a)(i)

Is the arc $\langle X, Y\rangle$ arc consistent?

Arc $\langle X, Y\rangle$ is arc consistent means that for every X there exists a Y such that $r(X, Y)$ is true.

The following table shows for each value of X a value of Y such that $R(X, Y)$ is true:

X	Y
a	d
b	a
c	***
d	i

Note that there is no element Y in $\{a, d, i, r\}$ such that $r(c, Y)$ is true. That is there is no word starting with *ca*, *cd*, *ci*, or *cr* in the word list. c can be removed from the domain of X to make $\langle X, Y\rangle$ arc consistent.

Solution to question 3(a)(ii)

Is the arc $\langle Y, X\rangle$ arc consistent?

Arc $\langle Y, X\rangle$ is arc consistent means that for every Y there exists an X such that $r(X, Y)$ is true.

Note that the relation $r(X, Y)$ corresponds to two arcs, $\langle X, Y\rangle$ and $\langle Y, X\rangle$. You do not swap the arguments to $r(X, Y)$. (This was the most common mistake).

The following table shows for each value of Y a value of X such that $R(X, Y)$ is true:

Y	X
a	b
d	a
i	d
r	a

Thus the arc $\langle Y, X\rangle$ is arc consistent.

Question 3 (b) & (c)

(b) In the STRIPS representation, predicates are divided into static, primitive and derived predicates. Use full sentences in your answer.

- **i)** [2 marks] What is a static predicate?
- **ii)** [2 marks] What is a primitive predicate?
- **iii)** [2 marks] What is a derived predicate?

(c) [3 marks] What is the object-attribute-value representation, and why is it of interest? Use full sentences to answer this question.

Solution to Question 3(b)

i) A static predicate is one the doesn't change through time.

ii) A primitive predicate is one whose truth value is derived from predicates at the previous time step.

iii) A derived predicate is one whose truth value is derived from predicates at the same time.

Solution to Question 3(b)

The object-attribute-value representation is where the is a single relation

$$prop(Obj, Att, Val)$$

which is true if object *Obj* have value *Val* for attribute *Att*.

It is of interest because it is the basis for flexible knowledge representation where any of the arguments can be queried.