CS322 Fall 1997
Practice Midterm Questions

Here are some questions for the midterm.

The following chapters/sections will be covered: Chapter 1, Chapter 2, Chapter 3 (except Section 3.7), Chapter 4, Chapter 5, Sections 7.2, 7.4, 8.1, and 8.2 (STRIPS representation only).

This practice midterm is longer than the actual midterm. It is designed to give you some impression of the types of questions that may be asked. You should also expect some questions based on the assignments.

Question 1

Given the knowledge base *KB* containing the clauses:

$a \leftarrow b \wedge c.$

$b \leftarrow d.$

$b \leftarrow e.$

$c.$

$d \leftarrow h.$

$e.$

$f \leftarrow g \wedge b.$

$g \leftarrow c \wedge k.$

$j \leftarrow a \wedge b.$

1. Show how the bottom-up proof procedure works for this example. Show at each step what element is added to *C*. Give all logical consequences of *KB*.
2. f isn't a logical consequence of *KB*. Explain what this means. Give a model of *KB* in which f is false.
3. a is a logical consequence of *KB*. Explain what this means. Give a top-down derivation for the query $?a$.

Solution to Question 1, part 1

The following shows the elements added to *C*

$\{c\}$

$\{e, c\}$
$\{e, c\}$
$\{b, e, c\}$
$\{a, b, e, c\}$
$\{j, a, b, e, c\}$

The set of logical consequences of *KB* is $\{j, a, b, e, c\}$.

Solution to Question 1, part 2

f isn't a logical consequence of *KB* means there is a model of the *KB* in which f is false.

One model of *KB* is where the atoms $\{j, a, b, e, c\}$ are true and all other atoms are false. f is false in this model.

Another model is where $\{h, d, j, a, b, e, c\}$ are all true, and all other atoms are false. f is false in this model too.

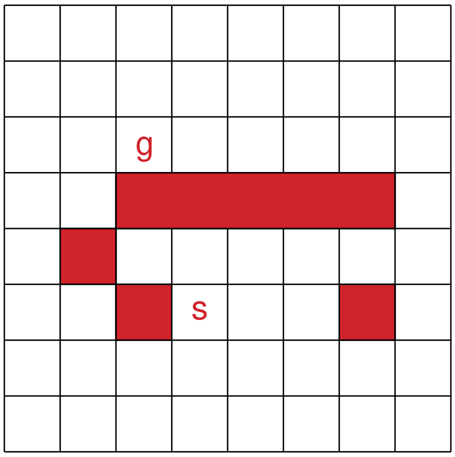
Solution to Question 1, part 3

a is a logical consequence of *KB* means that a is true in all models of *KB*. Here is a a top-down derivation for the query $?a$.

$yes \leftarrow a.$
$yes \leftarrow b \wedge c.$
$yes \leftarrow e \wedge c.$
$yes \leftarrow c.$
$yes \leftarrow .$

Question 2

Consider the problem of finding a path in the grid shown below from the position s to the position g. A piece can move on the grid horizontally and vertically, one square at a time. No step may be made into a forbidden shaded area.



1. On the grid, number the nodes in the order in which they are removed from the frontier in a depth-first search from s to g, given that the order of the operators you will test is: up, left, right, then down. Assume there is a cycle check.

2. Number the nodes in order in which they are taken off the frontier for an A^* search for the same graph. Manhattan distance should be used as the heuristic function. The Manhattan distance between two points is the distance in the x-direction plus the distance in the y-direction. It corresponds to the distance traveled along city streets arranged in a grid. For example, the Manhattan distance between g and s is 4. What is the path that is found?

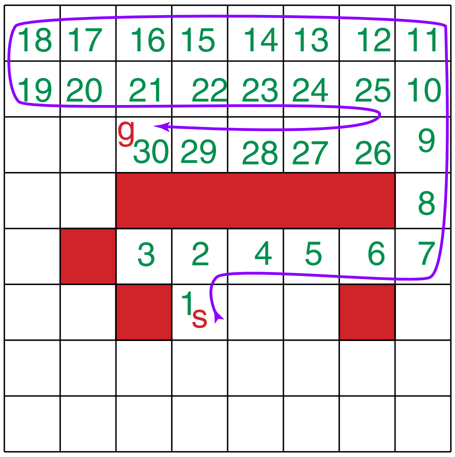
3. Assume that you were to solve the same problem using dynamic programming. Give the *dist* value for each node, and show which path is found.

4. Based on this experience, discuss which algorithms are best suited for this problem.

5. Suppose that the graph extended infinitely in all directions. That is, there is no boundary, but s, g, and the forbidden area are in the same relative positions to each other. Which methods would no longer find a path? Which would be the best method, and why?

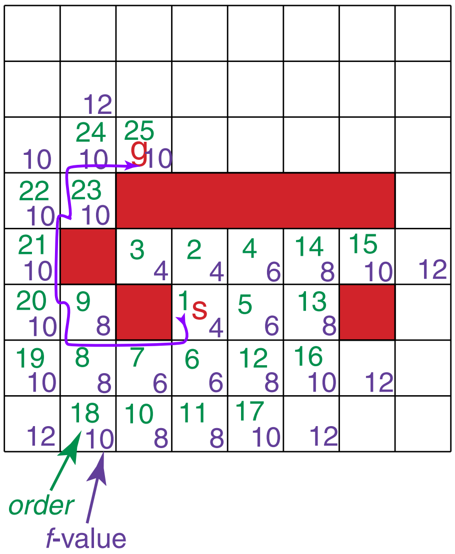
Solution to Question 2, part 1

Depth-first search:



Solution to Question 2, part 2

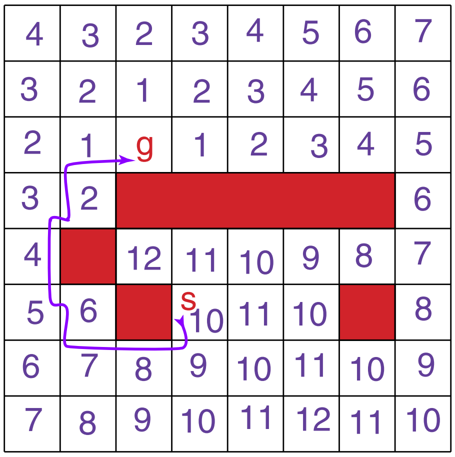
A^* search:



Note that here I assumed that for those nodes whose f-value is the same, that the last added to the queue is used.

Solution to Question 2, part 3

Dynamic Programming:



Solution to Question 2, part 4

It seems as though A^* and dynamic programming are best for this problem. A^* would be best if you were solving it once. If you wanted to solve it multiple times for the same goal dynamic programming would be good.

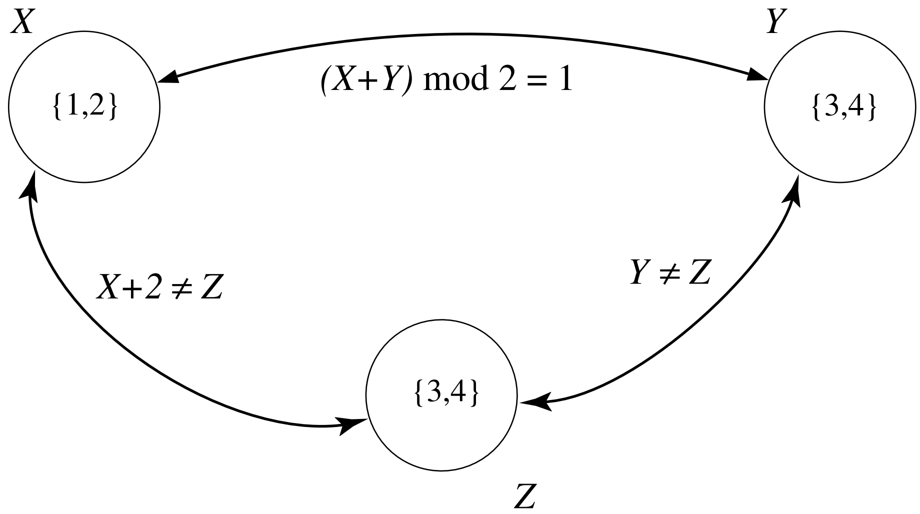
Depth-first search looks pretty stupid here. If you had searched in the order: left, up, right, down, it would have looked much more sensible.

Solution to Question 2, part 5

If the graph had extended infinitely in all directions, A^* would still find a solution. Depth-first search would wander off infinitely. Dynamic programming would find a path from s to g, but then continue off forever. If you stopped it when it had included s in the distance map, it would find the shortest path, but this would be the same as a shortest-first search from g, which still isn't as good as A^*. It would let us reuse the distance function build for other starting positions.

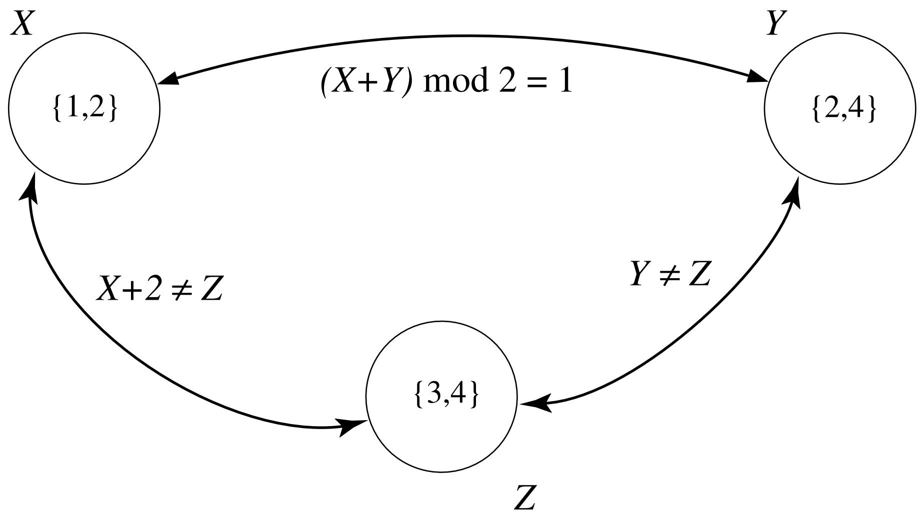
Question 3

1. Consider the following constraint graph. Note that $(X + Y) \bmod 2 = 1$ means that $X + Y$ is odd.



Is it arc consistent? If it is, explain why. If it isn't, explain which domain element can be pruned, and give the arc that can be used to prune it.

2. Consider the following constraint graph:



Is it arc consistent? If it is, explain why. If it isn't, explain which domain element can be pruned, and give the arc that can be used to prune it.

Solution to Question 3, part 1

Yes, this is arc consistent.

The relation, $(X+Y) \bmod 2 = 1$ is true of $X = 1, Y = 4$ and $X = 2, Y = 3$. For each X-value there is a Y-value and for each Y-value there is an X-value.

The relation $X + 2 \neq Z$ is true of $X = 1, Z = 4$ and $X = 2, Z = 3$. For each X-value there is a Z-value and for each Z-value there is an X-value.

The relation $Y \neq Z$ is true of $Y = 3, Z = 4$ and $Y = 4, Z = 3$. For each Y-value there is a Z-value and for each Z-value there is an Y-value.

Thus it is arc consistent, but there are no solutions!

Solution to Question 3, part 2

No, this is not arc consistent.

For $X = 2$ there is no Y such that $(X + Y) \bmod 2 = 1$ is true. This can be pruned with the arc $\langle X, Y\rangle$.

Question 4

Using the object-attribute-value representation, axiomatize the following knowledge about a user's preference for news articles using definite clauses (no negation as failure). The user can skip or read an article, and each article is either long or short, either a new thread or a follow-up, the author of the news article is either known or unknown to the user, and the article is either read at home or at work.

> The user skips long articles and short articles that are a follow-ups where the author is unknown.
> They read short articles whose thread is new or whose author is known.

There was a typo in the original version. The last word should be "known" (as above).

Note that this description covers all cases for articles.

Solution to Question 4

$$
\begin{aligned}
&\mathit{prop}(\mathit{Obj}, \mathit{user\_action}, \mathit{skips}) \leftarrow \\
&\quad \mathit{prop}(\mathit{Obj}, \mathit{length}, \mathit{long}). \\
&\mathit{prop}(\mathit{Obj}, \mathit{user\_action}, \mathit{reads}) \leftarrow \\
&\quad \mathit{prop}(\mathit{Obj}, \mathit{length}, \mathit{short}) \wedge \\
&\quad \mathit{prop}(\mathit{Obj}, \mathit{thread}, \mathit{new}). \\
&\mathit{prop}(\mathit{Obj}, \mathit{user\_action}, \mathit{reads}) \leftarrow \\
&\quad \mathit{prop}(\mathit{Obj}, \mathit{length}, \mathit{short}) \wedge \\
&\quad \mathit{prop}(\mathit{Obj}, \mathit{thread}, \mathit{follow\_up}) \wedge \\
&\quad \mathit{prop}(\mathit{Obj}, \mathit{author}, \mathit{known}). \\
&\mathit{prop}(\mathit{Obj}, \mathit{user\_action}, \mathit{skips}) \leftarrow \\
&\quad \mathit{prop}(\mathit{Obj}, \mathit{length}, \mathit{short}) \wedge \\
&\quad \mathit{prop}(\mathit{Obj}, \mathit{thread}, \mathit{follow\_up}) \wedge \\
&\quad \mathit{prop}(\mathit{Obj}, \mathit{author}, \mathit{unknown}).
\end{aligned}
$$

Question 5

Give Clark's completion of the following clauses for *live* and *connected_to*:

$$
\begin{aligned}
&live(Y) \leftarrow \\
&\quad connected\_to(Y, Z) \wedge \\
&\quad live(Z). \\
&live(outside). \\
&connected\_to(l_1, w_0). \\
&connected\_to(w_0, w_1) \leftarrow up(s_2). \\
&connected\_to(w_0, w_2) \leftarrow down(s_2).
\end{aligned}
$$

Solution to Question 5

$$
\begin{aligned}
&live(W) \leftrightarrow \\
&\quad (\exists Y\, \exists Z\; W = Y \wedge connected\_to(Y, Z) \wedge live(Z)) \\
&\quad \vee W = outside. \\
&connected\_to(X, Y) \leftrightarrow \\
&\quad (X = l_1 \wedge Y = w_0) \\
&\quad \vee (X = w_0 \wedge Y = w_1 \wedge up(s_2)) \\
&\quad \vee (X = w_0 \wedge Y = w_2 \wedge down(s_2)).
\end{aligned}
$$

Question 6

Using the domain of Section 8.1 and 8.2, give the STRIPS representation for the action $putdown(Ag, Obj)$. (Of course in the midterm, you will be given enough context so that you don't need to remember the example).

Solution to Question 6

$putdown(Ag, Obj)$:

Preconditions: $[carrying(Ag, Obj), at(Ag, Loc)]$

Delete List: $[carrying(Ag, Obj)]$

Add List: $[sitting\_at(Obj, Loc)]$