Please WAIT until we tell you to begin.

During the exam, you may refer to any written or printed paper material. **You may NOT use any electronic devices (including calculators, phones, etc).**

If you have questions, please come to us at the front to ask them.

**Enter all answers in the boxes provided.**

Extra work may be taken into account when assigning partial credit, but **only work shown on pages with QR codes will be considered.**

**Question 1:** 18 Points  
**Question 2:** 10 Points  
**Question 3:** 10 Points  
**Question 4:** 16 Points  
**Question 5:** 18 Points  
**Question 6:** 18 Points  
**Question 7:** 18 Points  
**Question 8:** 16 Points  
**Total:** 124 Points
1 Catching the Bus (18 Points)

Ben Bitdiddle needs to catch a bus in an infinite 2-d grid. We know the bus’s schedule, and we want to use the graph search algorithms we’ve built up in 6.01 to plan a path such that Ben and the bus end up in the same grid location at the same time.

Assume that the search starts at time 0 with Ben in location (0,0). On each step, time increases by 1, and Ben can move to any of the eight neighboring locations, or he can remain in the same location.

The position of the bus is specified by a procedure `bus_schedule`, which takes a time as input and returns the bus’s location at that time as a tuple \((r, c)\).

1.1 Paths

For the following implementations of the bus’s schedule, what is one sequence of locations (representing Ben’s location at time 0, 1, 2...) that might result from running a breadth-first search in this domain? Enter None if no path is returned, either because BFS runs indefinitely, or because it returns without having found a path. Note that there may be more than one correct answer, but you need only enter one.

1.1.1 Schedule 1

```python
def bus_schedule(t):
    return (1, t-2)
```

Path found by BFS: \([0,0], [1, -1]\)

1.1.2 Schedule 2

```python
def bus_schedule(t):
    return (2, (3*t) % 8)
```

Path found by BFS: \([0, 0], (1, -1), (2, 0), (2, 1)\)

1.1.3 Schedule 3

```python
def bus_schedule(t):
    return (t+1, t+1)
```

Path found by BFS: None (infinite loop)
1.2 Search Implementation

Recall that, in order to perform a search using the methods we have built up in 6.01, we need the following pieces:

- an appropriate successor function
- a goal test function
- a starting state

Enter your definitions for these pieces below.

```python
DIRECTIONS = [(0,0), (1,0), (0,1), (-1,0), (0,-1), (1,1), (1,-1), (-1,1), (-1,-1)]

def bus_catching_successors(state):
    r, c, time = state
    return [(r+dr, c+dc, time+1) for (dr, dc) in DIRECTIONS]

def goal_test(state):
    return state[:-1] == bus_schedule(state[-1])

start_state = (0,0,0)
```
3 Admissible (10 Points)

In this question, we will look at automating the process of deciding whether or not given heuristic functions are admissible in a particular search domain.

On the facing page, write a function admissible_heuristics that takes five arguments:

- `heuristics`: a list containing the heuristic functions to test
- `all_states`: a list containing all the states in the search space from which the goal is reachable
- `successors`: an appropriate successor function for searching in this domain using `uc_search`
- `goal_test`: a goal test function for searching in this space
- `path_cost`: a function that takes a path (a list of states) and returns the cost of that path

Your function should return a list containing only the heuristic functions from the `heuristics` list that are admissible. If none of the functions are admissible, it should return an empty list.

Note that the source code for the `uc_search` function from `lib601` is included for reference on the last page of this exam, which you may remove.
from lib601.search import uc_search

def admissible_heuristics(heuristics, all_states, successors, goal_test, path_cost):
    return [h for h in heuristics
            if is_admissible(h, all_states, successors, goal_test, path_cost)]

def is_admissible(h, all_states, successors, goal_test, path_cost):
    for start in all_states:
        if h(start) > path_cost(uc_search(successors, start, goal_test)):
            return False
    return True
5 Return of the Killer Hornets (18 Points)

Consider a world with some population $B$ of nice, pleasant bees and a population $H$ of crazy, evil, killer Japanese giant hornets. Bees don’t bother anyone, enjoying life and making honey. Hornets kill bees (when they can catch them). The bee population naturally increases, but when there are hornets around, they kill the bees and decrease the bee population. The hornet population, in the absence of honeybees to hunt, stays the same, or declines a little.

5.1 Initial distribution

Let’s assume that the bee population ($B$) can be low, med or high, and that the hornet population ($H$) can be low, med, or high. So, there are 9 states, each corresponding to some value of $B$ and some value of $H$.

Here is the initial belief state, which is written as a joint distribution over $B$ and $H$, $Pr(B, H)$.

<table>
<thead>
<tr>
<th>Bees</th>
<th>Hornets</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>low</td>
<td>med</td>
<td>high</td>
</tr>
<tr>
<td>low</td>
<td>0.04</td>
<td>0.20</td>
<td>0.18</td>
</tr>
<tr>
<td>med</td>
<td>0.08</td>
<td>0.16</td>
<td>0.02</td>
</tr>
<tr>
<td>high</td>
<td>0.28</td>
<td>0.04</td>
<td>0.00</td>
</tr>
</tbody>
</table>

1. What is the marginal distribution over the hornet population, $Pr(H)$?

$Pr(H) = (0.4, 0.4, 0.2)$

2. What is the distribution $Pr(B|H = \text{low})$?

$Pr(B|H = \text{low}) = (0.04/0.4, 0.08/0.4, 0.28/0.4) = (0.1, 0.2, 0.7)$

3. What is the distribution $Pr(B|H = \text{high})$?

$Pr(B|H = \text{high}) = (0.18/0.2, 0.02/0.2, 0.0/0.2) = (0.9, 0.1, 0.0)$


No, if they were independent, we would have, for example,

$Pr(B = \text{high}, H = \text{high}) = Pr(B = \text{high})P(H = \text{high})$

and it isn’t.
5.2 Transitions

Let’s start by studying how the bee population evolves when there are no hornets, represented by $H_t = \text{low}$ (and the hornet population doesn’t change).

- $\Pr(B_{t+1} = \text{low} \mid H_t = \text{low}, B_t = \text{low}) = 0.1$
- $\Pr(B_{t+1} = \text{med} \mid H_t = \text{low}, B_t = \text{low}) = 0.9$
- $\Pr(B_{t+1} = \text{high} \mid H_t = \text{low}, B_t = \text{low}) = 0.0$
- $\Pr(B_{t+1} = \text{low} \mid H_t = \text{low}, B_t = \text{med}) = 0.0$
- $\Pr(B_{t+1} = \text{med} \mid H_t = \text{low}, B_t = \text{med}) = 0.3$
- $\Pr(B_{t+1} = \text{high} \mid H_t = \text{low}, B_t = \text{med}) = 0.7$
- $\Pr(B_{t+1} = \text{low} \mid H_t = \text{low}, B_t = \text{high}) = 0.0$
- $\Pr(B_{t+1} = \text{med} \mid H_t = \text{low}, B_t = \text{high}) = 0.0$
- $\Pr(B_{t+1} = \text{high} \mid H_t = \text{low}, B_t = \text{high}) = 1.0$

1. Assume $H_t = \text{low}$. For simplicity, also assume that we start out knowing with certainty that the bee population is low. What is the distribution over the possible levels of the bee population (low, med, high) after 1 time step?

$\Pr(B_1 \mid B_0 = \text{low}, H_0 = \text{low}) = (0.1, 0.9, 0.0)$

2. Assume $H_t = \text{low}$. For simplicity, also assume that we start out knowing with certainty that the bee population is low. What is the distribution over the possible levels of the bee population (low, med, high) after 2 time steps?

$\Pr(B_2 \mid B_0 = \text{low}, H_0 = \text{low}) = (0.01, 0.09 + 0.24, 0.63) = (0.01, 0.36, 0.63)$

3. Assume $H_t = \text{low}$. For simplicity, also assume that we start out knowing with certainty that the bee population is low. What value does the distribution over the possible levels of the bee population (low, med, high) approach as the number of time steps goes to infinity?

$\Pr(B_{\infty} \mid B_0 = \text{low}, H_0 = \text{low}) = (0, 0.1)$
5.3 Observations

Imagine that you are starting with the initial belief state, as given by the joint distribution from problem 5.1. If it helps, you can think of it as the following $\text{DDist}$ over pairs of values (the first is the value of $B$, the second is the value of $H$):

$$\text{DDist}({('low', 'low') : 0.04, ('low', 'med') : 0.2, ('low', 'high') : 0.18, ('med', 'low') : 0.08, ('med', 'med') : 0.16, ('med', 'high') : 0.02, ('high', 'low') : 0.28, ('high', 'med') : 0.04, ('high', 'high') : 0.00})$$

You send an ecologist out into the field to sample the numbers of bees and hornets. The ecologist can’t really figure out the absolute numbers of each species, but reports one of three observations:

- **moreH**: means that there are significantly more hornets than bees (that is, that the level of hornets is $\text{high}$ and the level of bees is $\text{med}$ or $\text{low}$, or that the level of hornets is $\text{med}$ and the level of bees is $\text{low}$).

- **moreB**: means that there are significantly more bees than hornets (that is, that the level of bees is $\text{high}$ and the level of hornets is $\text{med}$ or $\text{low}$, or that the level of bees is $\text{med}$ and the level of hornets is $\text{low}$).

- **same**: means that there are roughly the same number of hornets as bees (the populations have the same level).

1. If there is no noise in the ecologist’s observations (that is, the observation is always true, given the state), and the observation is **moreB**, what is the resulting belief state $\Pr(B_0, H_0 \mid O_0 = \text{moreB})$ (the distribution over states given the observation)? Fill in the table below with the joint probabilities in the updated belief:

<table>
<thead>
<tr>
<th></th>
<th>low</th>
<th>med</th>
<th>high</th>
</tr>
</thead>
<tbody>
<tr>
<td>low</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Bees</td>
<td>med</td>
<td>0.2</td>
<td>0</td>
</tr>
<tr>
<td>high</td>
<td>0.7</td>
<td>0.1</td>
<td>0</td>
</tr>
</tbody>
</table>
2. Now, we will assume that the ecologist’s observations are fallible.

\[
\begin{align*}
\Pr(O_t = \text{more}H | B_t < H_t) &= 0.9 \\
\Pr(O_t = \text{same} | B_t < H_t) &= 0.1 \\
\Pr(O_t = \text{more}B | B_t < H_t) &= 0.0 \\
\Pr(O_t = \text{more}H | B_t = H_t) &= 0.1 \\
\Pr(O_t = \text{same} | B_t = H_t) &= 0.8 \\
\Pr(O_t = \text{more}B | B_t = H_t) &= 0.1 \\
\Pr(O_t = \text{more}H | B_t > H_t) &= 0.0 \\
\Pr(O_t = \text{same} | B_t > H_t) &= 0.1 \\
\Pr(O_t = \text{more}B | B_t > H_t) &= 0.9
\end{align*}
\]

Again starting from the initial belief state, if the observation is \textbf{moreB}, what is the belief state \( \Pr(B_0, H_0 | O_0 = \text{more}B) \) (the distribution over states given the observation)? Fill in the table below with the joint probabilities in the updated belief:

<table>
<thead>
<tr>
<th></th>
<th>Hornets</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>low</td>
<td>med</td>
<td>high</td>
</tr>
<tr>
<td>low</td>
<td>4/380</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Bees</td>
<td>72/380</td>
<td>16/380</td>
<td>0</td>
</tr>
<tr>
<td>high</td>
<td>252/380</td>
<td>36/380</td>
<td>0</td>
</tr>
</tbody>
</table>
8 Say "Cheese!" (16 Points)

The workers at a cheese shop throw completed wheels of cheese into a barrel that holds 4 cheese wheels, and customers come and buy cheese wheels from the barrel. The shop is still a small-time operation, so they only have two workers.

At the beginning of each hour (at $x:01$), each worker makes a new cheese wheel and adds it to the barrel with a probability that depends on the number $n$ of cheese wheels already in the barrel:

$$
\Pr(\text{make a new cheese wheel}) = \begin{cases} 
1/2 & \text{if } n = 0 \\
1/2 & \text{if } n = 1 \\
1/4 & \text{if } n = 2 \\
0 & \text{otherwise}
\end{cases}
$$

You can assume that each hour, the two workers operate based on the same value of $n$ (measured at $x:00$). You may also assume that, like all good workers, they never take breaks, they never sleep, they will live forever, and they will never retire.

At the end of each hour (at $x:59$), each cheese wheel in the barrel is sold (and thus is removed from the barrel) with probability $1/2$, independently of whether the other wheels sell.

For each of the questions below, please enter a single fraction in the box, representing the probability in question.

1. At 10:30am on Monday, your belief over the number of cheese wheels in the barrel is given by:

   $\text{DDist}(\{1: \frac{1}{3}, 3: \frac{2}{3}\})$

   What is the probability that there is exactly one cheese wheel left in the barrel at 11:00am?

   \[
   \frac{5}{12}
   \]

2. At 4:30pm on Tuesday, you are told that there are exactly 2 wheels of cheese in the barrel. What is the probability that neither worker will make a wheel of cheese at 5:01pm?

   \[
   \frac{21}{64}
   \]
3. At 1:30pm on Wednesday, you have no idea how many cheese wheels are in the barrel (it is equally likely that there are 0, 1, 2, 3, 4 cheese wheels). You then see that exactly 3 cheese wheels sell at 1:59pm. What is the probability that there is exactly 1 cheese wheel left in the barrel?

\[
\frac{2}{3}
\]

4. At 3:30pm on Thursday, you know that there are exactly 2 cheese wheels in the barrel. At 4:30pm, there are exactly 3 cheese wheels in the barrel. What is the probability that both workers made a cheese wheel at 4:01pm?

\[
\frac{4}{7}
\]

5. At 12:30pm on Friday, you look and see that there is exactly 1 cheese wheel in the barrel. Coming back at 2:30pm (2 hours later), you are told that no cheese sold in the last two hours, and you notice that there are exactly 3 cheese wheels in the barrel. What is the probability that the same worker made both of the new wheels of cheese?

\[
\frac{3}{16}
\]
1 Decodence (11 Points)

1.1 Sender
Consider the following system, designed to take in an input signal, and to output a different, coded form of the signal for transmission.

\[ X_s \xrightarrow{\mathcal{R}} \alpha \xrightarrow{+} Y_s \]

For what values of \( \alpha \) is this system stable? Enter an equation or inequality in the box below:

\[ -\infty < \alpha < \infty \]

1.2 Receiver
Consider also this system, which is designed to try to decode the signal from the sender.

\[ X_r \xrightarrow{+} \mathcal{R} \xrightarrow{\beta} Y_r \]

For what values of \( \beta \) is this system stable? Enter an equation or inequality in the box below:

\[ |\beta| \leq 1 \]
1.3 Combination

Now consider the case when the two systems are connected in cascade (so that \( Y_s = X_r \)).

For what values of \( \alpha \) and \( \beta \) is this system stable? Enter equations and/or inequalities in the box below:

\[
|\alpha| < \infty, |\beta| \leq 1
\]

Assume a fixed, nonzero value of \( \alpha \). For what values of \( \beta \) is the unit sample response of the system a delayed unit sample signal? Enter equations and/or inequalities in the box below, or enter None if no value of \( \beta \) would cause this behavior:

\[ \beta = -\alpha \]

Under the conditions from the previous question, what would be the output of the system when the input is the sequence \([1, 3, 5, 6, 7, 0, 0, 0, \ldots]\)? Calculate the first 6 samples of the systems output in this case, and enter them in the box below:

\([0, 1, 3, 5, 6, 7]\)
4 Hot Bath (6 points)

A thermocouple is a physical device with two temperature probes and two electronic terminals. If the probes are put in locations with different temperatures, there will be a voltage difference across the terminals. In particular,

\[ V_+ - V_- = k(T_h - T_r) \]

where \( T_h \) is the temperature (°F) at the ‘hot’ probe, \( T_r \) is the temperature (°F) at the reference probe, and \( k \) is about 0.02.

We have a vat of liquid that contains a heater (the coil at the bottom) and the ‘hot’ temperature sensor of the thermocouple. We would like to keep the liquid at the same temperature as the reference probe. The heater should be off if \( T_r \leq T_h \) and be on, otherwise. When \( T_r - T_h = 1 \) °F, then \( V_O \) should be approximately +5V.

Design a simple circuit (using one or two op-amps and some resistors of any values you want) that will achieve this; pick particular values for the resistors. Assume that we have a power supply of +10V available, and that the voltage difference \( V_+ - V_- \) is in the range -10V to +10V.