

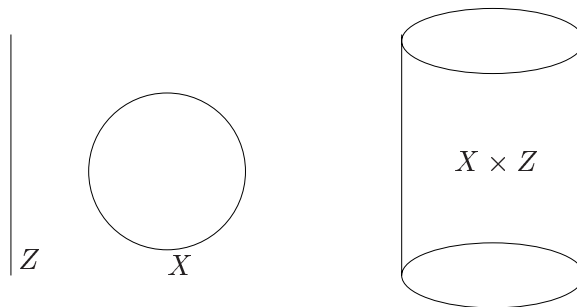
EECS 20. Practice Problems No. 1.

You have the following sources of practice problems:

- Quizzes you have taken.
- Problem sets you have turned in.
- The exam archive on the web from prior semesters.
- The labs you have done.
- The problem session problems.
- The following problems, mostly drawn from the problem sessions.

In the problems below, references to figures and sections are references to your reader, except when the figure is included here.

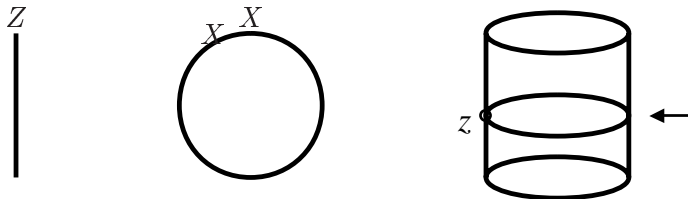
A.4 **T** In the following figure, $X = \{(x, y) \mid x^2 + y^2 = 1\}$ is depicted as a 2-dimensional circle and $Z = [0, 2]$ is shown as a 1-dimensional line segment. Explain why it is reasonable to show the product set as a 3-dimensional cylinder.



Solution: The product of an m -dimensional set with an n -dimensional set is a new set with $m + n$ dimensions. One way to think of $Z \times X$ is to imagine that each point in the line becomes a circle. That is, each and every element in the 1-dimensional line is combined with every element in the 2-dimensional circle set. The new 3-dimensional set, when graphed, is a cylinder. In symbols,

$$Z \times X = \bigcup_{z \in Z} \{z\} \times X$$

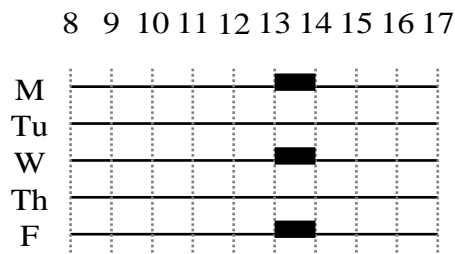
Graphically, the product can be drawn as:



The arrow points to $\{z\} \times X$.

A.5 **E** In the spirit of figure A.2, give a picture for the product set $\{M, Tu, W, Th, F\} \times [8.00, 17.00]$ and indicate on your drawing the lecture hours for this class.

Solution: Sketch:



The above assumes the class is scheduled to meet MWF 11-12.

A.6 **E** In the spirit of figure A.2, give a picture for the set $A = \{(x, y) \mid x \in [1, 2], y \in [1, 2]\}$ and the set $B = \{(x, x) \mid x \in [1, 2]\}$. Explain why the two sets are different.

Solution: Sketch:



In the second set, both elements of the tuple must be identical. In the first set, they can be different.

A.13 **E** Define the following sets in terms of the sets named in section A.1.5.

- (a) The set of all 10-letter passwords.
- (b) The set of all 5×6 matrices of real numbers.

- (c) The set of all complex numbers with magnitude at most 1.
- (d) The set of all 2-dimensional vectors with magnitude exactly 1.

Solution:

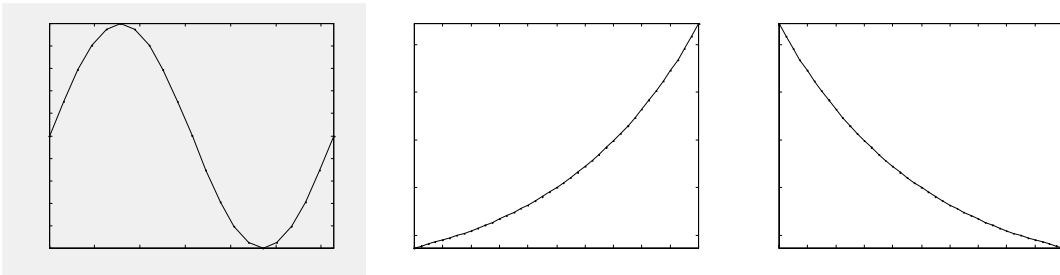
- (a) \mathbb{C}^{10}
- (b) \mathbb{R}^{30} or $(\mathbb{R}^{5})^6$ or $(\mathbb{R}^6)^5$.
- (c) $\{c \in \mathbb{C} \mid |c| \leq 1\}$.
- (d) $\{(v_1, v_2) \in \mathbb{R}^2 \mid v_1^2 + v_2^2 = 1\}$.

A.16 **T** Use Matlab to depict the following sets using the plot command:

- (a) $\{(t, x) \in \mathbb{R}^2 \mid x = \sin(t), \text{ and } t \in \{0, \frac{1}{20}2\pi, \frac{2}{20}2\pi, \dots, \frac{20}{20}2\pi\}\}$,
- (b) $\{(y, x) \in \mathbb{R}^2 \mid y = e^x, \text{ and } x \in \{-1, -1 + \frac{1}{20}, -1 + \frac{2}{20}, \dots, 1\}\}$,
- (c) $\{(y, x) \in \mathbb{R}^2 \mid y = e^{-x}, \text{ and } x \in \{-1, -1 + \frac{1}{20}, -1 + \frac{2}{20}, \dots, 1\}\}$.

Solution:

- (a) `t=0:(pi/10):(2*pi); plot(t,sin(t));`
- (b) `x=-1:(1/20):1; plot(x,exp(x));`
- (c) `x=-1:(1/20):1; plot(x,exp(-x));`



1.1 **C** For each of the continuous-time signals below, represent the signal in the form of $f: X \rightarrow Y$ and as a sketch in the form of figure 1.1. Carefully identify the range and domain in each case.

- (a) The voltage across a car battery,
- (b) The closing prices on each day of a share of a company,
- (c) The position of a moving vehicle on a straight road,
- (d) The simultaneous position of two moving vehicles on the same straight road, and
- (e) The sound heard in both of your ears.

Solution:

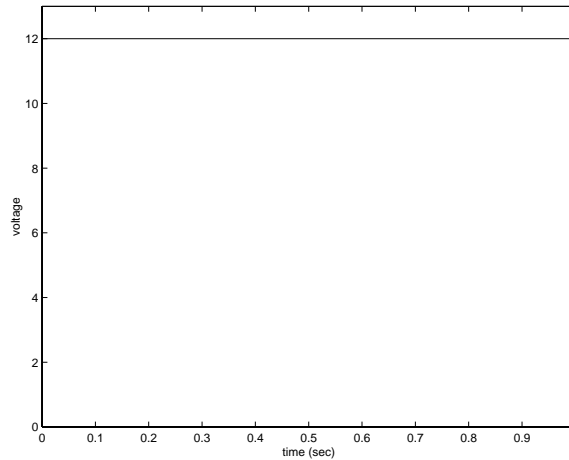
(a) One possible model is

$$\text{CarBatteryVoltage}: \text{Reals} \rightarrow \text{Reals}$$

where the domain represents time and the range represents voltage, and

$$\forall t \in \text{Reals}, \quad \text{CarBatteryVoltage}(t) = 12.$$

Since the value is constant, we could have defined the range to be $\{12\}$, a set with just one element. However, in practice, the voltage is only approximately a constant 12 volts. In fact, while you are starting your car, the electrical load on the battery causes the voltage to drop considerably. The voltage also fluctuates with noise caused by electrical circuits in the car. Thus, setting the range to *Reals* gives us more modeling flexibility, allowing us to be more accurate if we choose to do so. A sketch is shown below:



This is generated by the following Matlab commands

```
plot([0 1],[12 12])
xlabel('time (sec)')
ylabel('voltage')
axis([0 1 0 13])
```

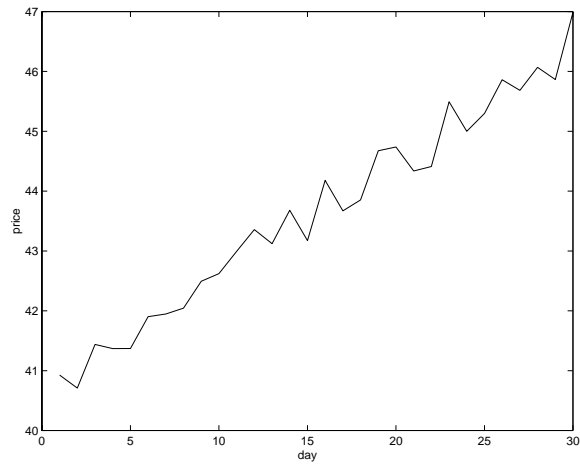
(b) One possible model is

$$\text{price}: \text{Nats} \rightarrow \text{Prices}$$

where the domain counts off days from some starting point, and the set *Prices* is

$$\text{Prices} = \{0, 1/16, 1/8, 3/16, 2/8, 5/16, \dots\},$$

assuming the stock markets have not yet switched a decimal system. A sketch of a segment of one such possible function (for a fictitious company) is shown below:



This is generated by the following Matlab commands

```

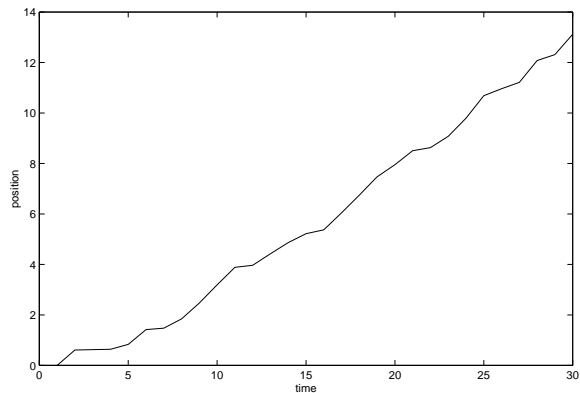
days = 1:30;
for day = days
    price(day) = 40 + day*0.2 + rand(1);
end
plot(days, price), xlabel('day'), ylabel('price')

```

(c) One possible model is

$$position: Reals \rightarrow [0, L]$$

where the domain represents time and L is the length of the road. A sketch of a segment of one such possible function (which assumes that the car is moving forward in fits and starts) is shown below:



This is generated by the following Matlab commands

```

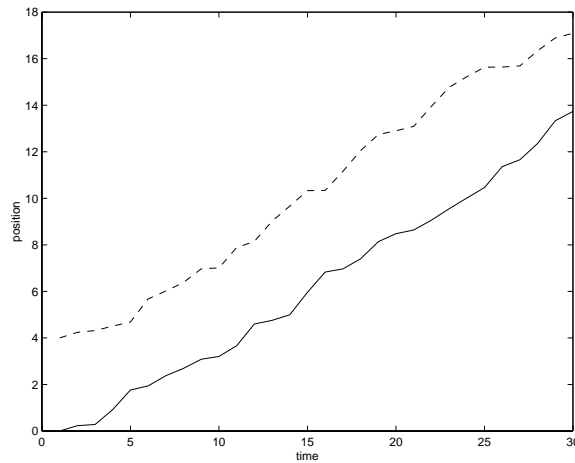
numSamples = 30;
times = 2:numSamples;
position(1) = 0;
for t = times
    position(t) = position(t-1) + rand(1);
end
plot([1 times], position), xlabel('time'), ylabel('position')

```

(d) One possible model is

$$twoPositions: Reals \rightarrow [0, L] \times [0, L]$$

where the domain represents time and L is the length of the road. A sketch of a segment of one such possible function (which assumes that both cars are moving forward in fits and starts) is shown below:



Here, the function is represented as two functions,

$$onePosition: Reals \rightarrow [0, L],$$

$$anotherPosition: Reals \rightarrow [0, L],$$

such that

$$\forall t \in Reals, \quad twoPositions(t) = (onePosition(t), anotherPosition(t)).$$

These are both plotted on the same plot, although it would equally valid to plot them in separate plots. This is generated by the following Matlab commands

```

numSamples = 30;
times = 2:numSamples;
position1(1) = 0;
position2(1) = 4;
for t = times
    position1(t) = position1(t-1) + rand(1);
    position2(t) = position2(t-1) + rand(1);
end

```

```
plot([1 times], position1, '-', [1 times], position2, '--')
xlabel('time'), ylabel('position')
```

(e) One possible model is

$$\text{stereoAudio}: \text{Reals} \rightarrow \text{Reals} \times \text{Reals}$$

where the domain represents time and the range represents air pressure at each of the two ears. Just as with the previous part, we can divide the function into two and plot the two functions separately or on the same plot.

1.4 C Formulate the following items of information as functions. Identify the domain and range in each case.

- (a) The population of U.S. cities,
- (b) The white pages in a phone book (careful: the white pages may list two identical names, and may list the same phone number under two different names),
- (c) The birth dates of students in class,
- (d) The broadcast frequencies of AM radio stations,
- (e) The broadcast frequencies of FM radio stations, (look at your radio dial, or at the web page:

<http://www.eecs.berkeley.edu/~eal/eecs20/sidebar/radio/index.html>.

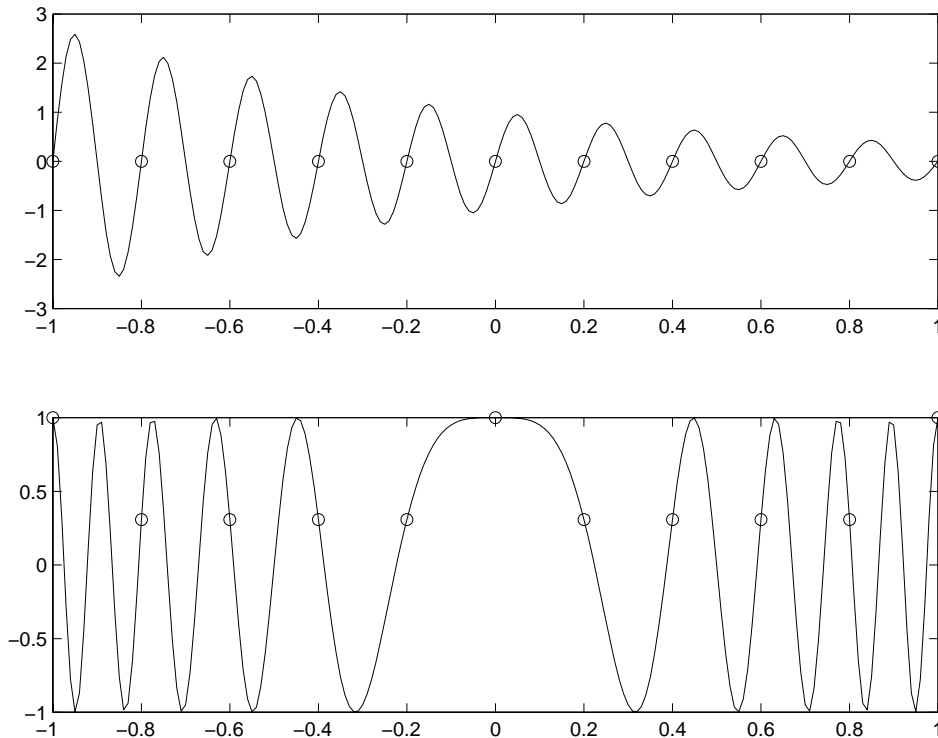
Solution:

- (a) $\text{Population}: \text{Cities} \rightarrow \text{Nats}$
where *Cities* is the set of cities.
- (b) This problem is a little tricky, since it is possible for there to be two entries in the white pages with the same name, and two entries with the same phone number. Thus, it will not work to map names into phone numbers, nor phone numbers into names. Instead, we choose the domain *Nats* to represent the position of the white pages entry in the phone book, and the range to be $\text{Names} \times \{0, 1, \dots, 9\}^7$.
- (c) $\text{Birthdates}: \text{Students} \rightarrow \text{Dates}$
where the sets are obvious.
- (d) $\text{Frequencies}: \text{Stations} \rightarrow [530, 1700]$
where *Stations* = KDOG, ... and the frequencies are in units of kHz. Note that we could actually be more specific and make the range a discrete set, since the FCC only allocates certain discrete frequencies.
- (e) $\text{Frequencies}: \text{Stations} \rightarrow [88, 108]$
where *Stations* = KQED, KPFA, ... and the frequencies are in units of MHz. Note that again we could actually be more specific and make the range a discrete set, since the FCC only allocates certain discrete frequencies.

1.5 E Use Matlab to plot the graph of the following continuous-time functions defined over $[-1, 1]$, and on the same plot display 11 uniformly spaced samples (0.2 seconds apart) of these functions.

- (a) $f: [-1, 1] \rightarrow \text{Reals}$, where for all $x \in [-1, 1]$, $f(x) = e^{-x} \sin(10\pi x)$.
 (b) *Chirp*: $[-1, 1] \rightarrow \text{Reals}$, where for all $t \in [-1, 1]$, $\text{Chirp}(t) = \cos(10\pi t^2)$.

Solution: The two plots are shown below:



Notice that in the first case, the samples all have value 0. In the second case, the samples also do not accurately reflect the structure of the waveform. This is a clear indication that we have not sampled frequently enough. The Matlab commands that generated these plots are:

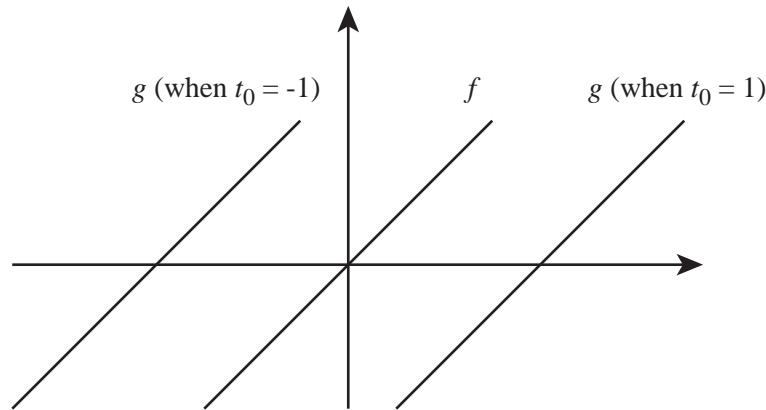
```
denseTime = [-1:0.01:1];
sampleTime = [-1:0.2:1];
y1 = exp(-denseTime).*sin(10*pi*denseTime);
y2 = exp(-sampleTime).*sin(10*pi*sampleTime);
subplot(2,1,1), plot(denseTime,y1,'-',sampleTime,y2,'o');
chirp1 = cos(10*pi*denseTime.^2);
chirp2 = cos(10*pi*sampleTime.^2);
subplot(2,1,2), plot(denseTime,chirp1,'-',sampleTime,chirp2,'o');
```

2.2 **T** This problem studies the relationship between the notion of **delay** and the graph of a function.

- (a) Consider two functions f and g from *Reals* into *Reals* where $\forall t \in \text{Reals}, f(t) = t$ and $g(t) = f(t - t_0)$, where t_0 is a fixed number. Sketch a plot of f and g for $t_0 = 1$ and $t_0 = -1$. Observe that if $t_0 > 0$ then $\text{graph}(g)$ is obtained by moving $\text{graph}(f)$ to the right, and if $t_0 < 0$ by moving it to the left.
- (b) Show that if $f: \text{Reals} \rightarrow \text{Reals}$ is any function whatsoever, and $\forall t, g(t) = f(t - t_0)$, then if $(t, y) \in \text{graph}(f)$, then $(t + t_0, y) \in \text{graph}(g)$. This is another way of saying that if $t_0 > 0$ then the graph is moved to the right, and if $t_0 < 0$ then the graph is moved to the left.
- (c) If t represents time, and if $t_0 > 0$, we say that g is obtained by *delaying* f . Why is it reasonable to say this?

Solution:

- (a) The functions are shown below:



- (b) If $(t, y) \in \text{graph}(f)$ then $f(t) = y$. Hence $g(t + t_0) = f((t + t_0) - t_0) = f(t) = y$, so $(t + t_0, y) \in \text{graph}(g)$.
- (c) If a particular value y occurs for f at time t , then the same value y will occur for g at time $t + t_0$. For $t_0 > 0$ this is later in time, so it is reasonable to call this a delay.
1. **C** A router in the Internet is a switch with several input ports and several output ports. A packet containing data arrives at an input port at an arbitrary time, and the switch forwards the packet to one of the outgoing ports. The ports of different routers are connected by transmission links. When a packet arrives at an input port, the switch examines the packet, extracting from it a destination address d . The switch then looks up the output port in its routing table, which contains entries of the form $(d, \text{outputPort})$. It then forwards the packet to the specified output port. Thus, the internet works by setting up the routing tables in the routers.

Consider a simplified router with one input port and two output ports, named O_1, O_2 . Let D be the set of destination addresses.

- (a) Explain why the routing table can be described as a subset $T \subset D \times \{O_1, O_2\}$.
- (b) Is it reasonable to constrain T to be the graph of a function from $D \rightarrow \{O_1, O_2\}$? Why?

- (c) Assume the signal at the input port is a sequence of packets. How would you describe the space of input signals to the router and output signals from the router?
- (d) How would you describe the switch as a function from the space of input signals to the space of output signals?

Solution:

- (a) Each entry in the table T is of the form $(d, outputPort)$, where $d \in D$ and $outputPort \in \{O_1, O_2\}$. Thus, the entry $(d, outputPort) \in D \times \{O_1, O_2\}$. Hence, $T \subset D \times \{O_1, O_2\}$.
- (b) T should be the graph of a function because for each destination address there should be listed exactly one output port, or else the router would not know what to do with the packet.
- (c) Let $Packets = D \times otherData$ be the set of possible packets. The set $otherData$ is the set of possible data, other than destination addresses, that the router carries. Define

$$PacketStream: Nats \rightarrow Packets.$$

Then an input to the router is in $PacketStream$, and an output is in $PacketStream^2$.

- (d) The switch is a function

$$Switch: PacketStream \rightarrow PacketStream^2$$

such that if the input $x \in PacketStream$ and $Switch(x) = (y, z)$, then y is the substream of x whose destination addresses satisfy $T(d) = O_1$ and z is the substream of x whose destination addresses satisfy $T(d) = O_2$.

2.11 E Suppose that $f: Reals \rightarrow Reals$ and $g: Reals \rightarrow Ints$ such that for all $x \in Reals$,

$$g(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases}$$

and

$$f(x) = 1 + x.$$

- (a) Define $h = g \circ f$.
- (b) Suppose that

$$\begin{aligned} F: [Reals \rightarrow Reals] &\rightarrow [Reals \rightarrow Reals] \\ G: [Reals \rightarrow Reals] &\rightarrow [Reals \rightarrow Ints] \end{aligned}$$

such that for all $s \in [Reals \rightarrow Reals]$ and $x \in Reals$,

$$\begin{aligned} (F(s))(x) &= f(s(x)) \\ (G(s))(x) &= g(s(x)) \end{aligned}$$

where f and g are as given above. Sketch a block diagram for $H = G \circ F$. Let $s \in [Reals \rightarrow Reals]$ be such that for all $x \in Reals$,

$$s(x) = \cos(\pi x).$$

Define u where

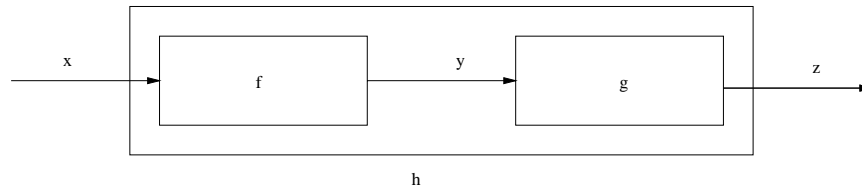
$$u = (G \circ F)(s).$$

Solution:

(a) $h: \text{Reals} \rightarrow \text{Ints}$ such that for all $x \in \text{Reals}$,

$$h(x) = \begin{cases} 1 & \text{if } x > -1 \\ 0 & \text{if } x = -1 \\ -1 & \text{if } x < -1 \end{cases}$$

(b) Here is a block diagram:



The desired signal is $u: \text{Reals} \rightarrow \text{Ints}$ where for all $x \in \text{Reals}$,

$$u(x) = \begin{cases} 0 & \text{if } x \text{ is an odd integer} \\ 1 & \text{otherwise} \end{cases}$$

Note that this signal is 1 almost everywhere and goes to zero only a countable number of discrete points on the real line. No real-world signal would behave this way.

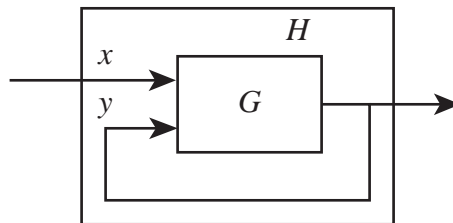
2.12 **T** Let $D = \text{DiscSignals} = [\text{Ints} \rightarrow \text{Reals}]$ and let

$$G: D \times D \rightarrow D$$

such that for all $x, y \in D$ and for all $n \in \text{Ints}$,

$$(G(x, y))(n) = x(n) - y(n - 1).$$

Now suppose we construct a new system H as follows:



Define H (as much as you can).

Solution: $H: D \rightarrow D$ such that for all $x \in D$ and $n \in \text{Ints}$, $H(x)$ satisfies

$$(H(x))(n) = x(n) - (H(x))(n - 1).$$

This is as much as we can define H . This definition is complete only if such $H(x)$ exists and is unique.

3.3 **E** Consider the alphabets

$$Inputs = Outputs = Bin = \{0, 1\}.$$

- (a) Construct a state machine that uses these alphabets such that if $x(0), x(1), \dots$ is any input sequence, the output sequence is

$$\forall n \in \mathbb{N}_{\geq 0}, \quad y(n) = \begin{cases} 1 & \text{if } n \geq 2 \wedge (x(n-2), x(n-1), x(n)) = (1, 1, 1) \\ 0 & \text{otherwise} \end{cases}$$

In words, the machine outputs 1 if the three previous inputs are all 1's, otherwise it outputs 0.

- (b) For the same input and output alphabet, construct a state machine that outputs 1 if the three previous inputs are either $(1, 1, 1)$ or $(1, 0, 1)$, and otherwise it outputs 0.

Solution: The solution is shown in figure 1. For part (a), $States = \{0, 1, 11\}$ while for part (b), $States = \{0, 1, 11, 10\}$.

3.6 **T** Construct an infinite state machine that realizes *Equal*.

Solution: From the description of *Equal*, it is clear that

$$Inputs = \{0, 1, absent\},$$

and

$$Outputs = \{equal, notEqual, absent\}.$$

We can define the state to be difference between the number of ones that have arrived and the number of zeros that have arrived, in which case,

$$States = Ints.$$

Then for all $s \in States$ and $x \in Inputs$,

$$update(s, x) = \begin{cases} (0, equal) & \text{if } (x = 1 \wedge s = -1) \vee (x = 0 \wedge s = 1) \\ (s + 1, notEqual) & \text{if } x = 1 \wedge s \neq -1 \\ (s - 1, notEqual) & \text{if } x = 0 \wedge s \neq 1 \\ (s, absent) & \text{otherwise} \end{cases}$$

- 3.10 **E** Consider the state machine in figure 3. It implements *CodeRecognizer*, but has more states than the one in figure 2. Show that it is equivalent by giving a bisimulation relation with the machine in figure 2.

Solution: The bisimulation relation is

$$\{(start, start), (1, 1), (11, 11), (110, 110), (start, 1100)\}.$$

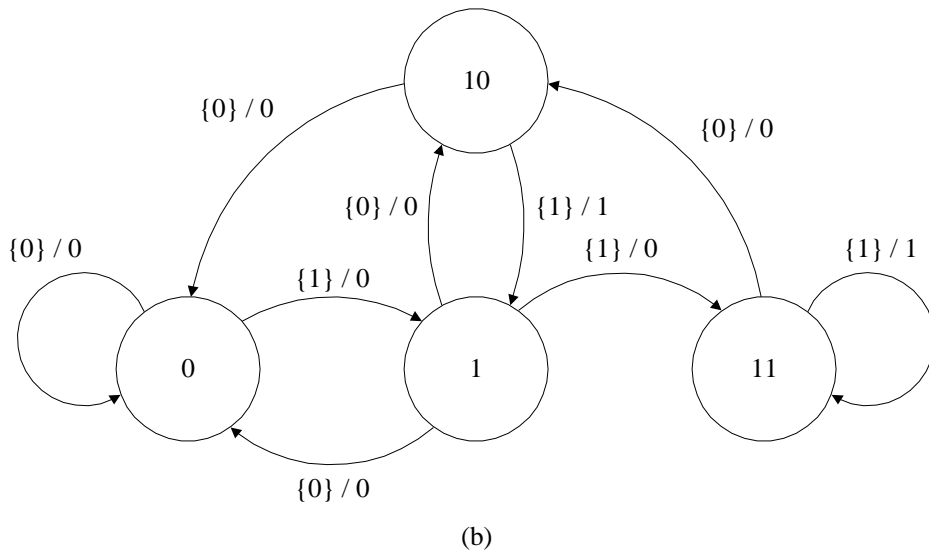
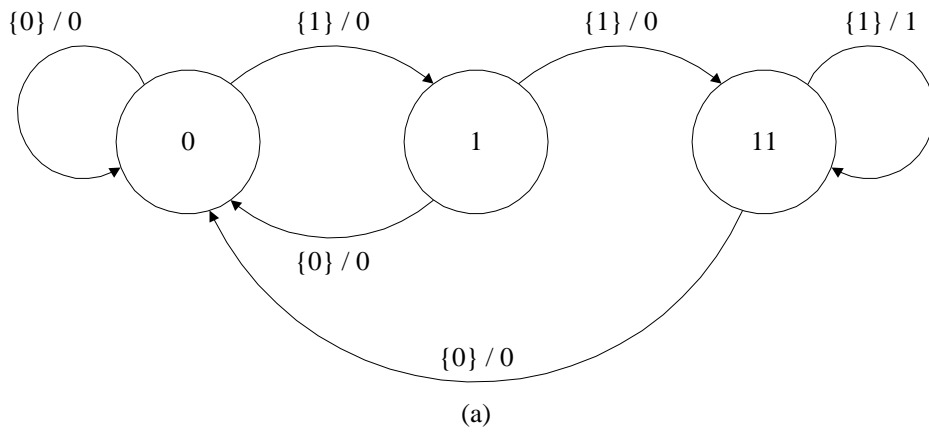


Figure 1: Solution to exercise 1.

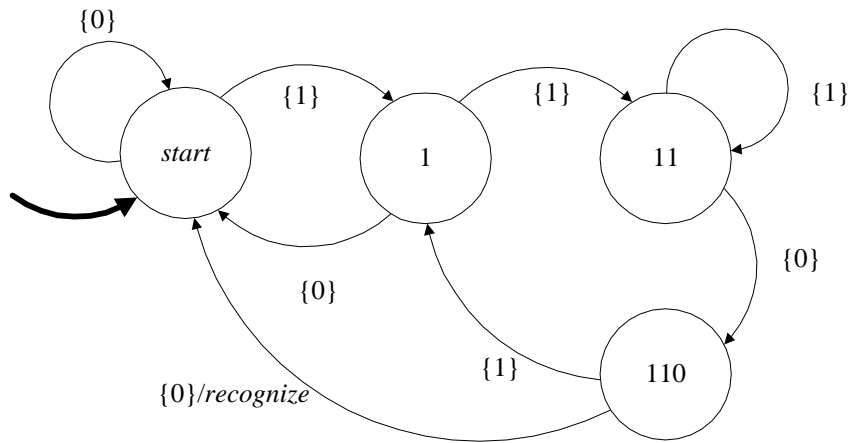


Figure 2: A machine that implements *CodeRecognizer*

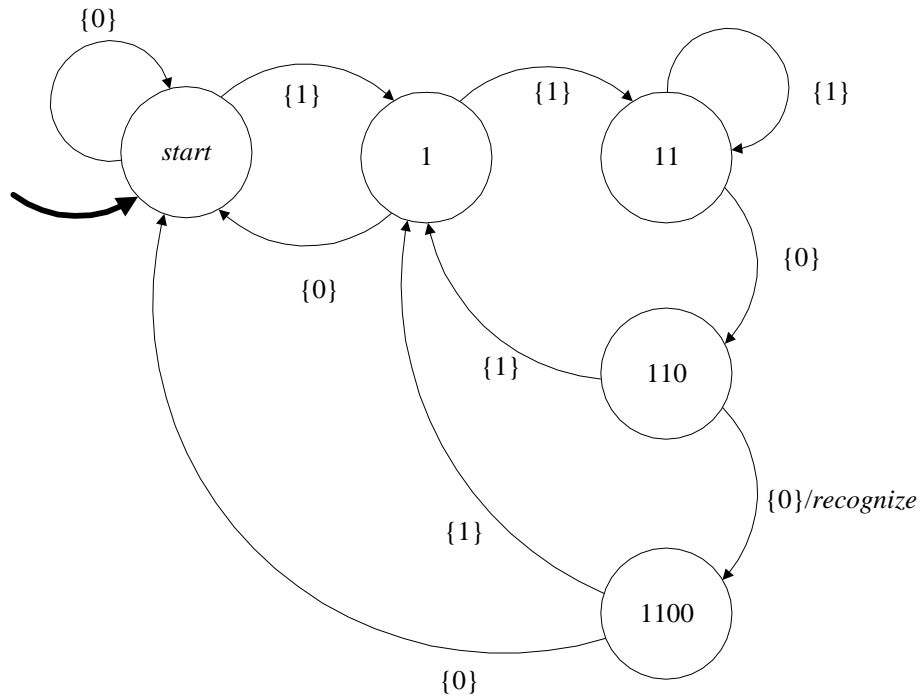


Figure 3: A machine that implements *CodeRecognizer*, but has more states than the one in figure 2.

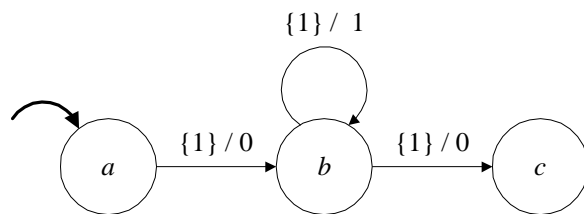


Figure 4: State machine for problem 1

3.11 E Consider the state machine in figure 4. Suppose that the alphabets are

$$\text{Inputs} = \{1, a\}$$

$$\text{Outputs} = \{0, 1, a\},$$

where a (short for *absent*) is the stuttering element. State whether each of the following is in the set *Behaviors* for this machine. In each of the following, the ellipsis “ \dots ” means that the last element is repeated forever. Also, in each case, the input and output signals are given as sequences.

- (a) $((1, 1, 1, 1, 1, \dots), (0, 1, 1, 0, 0, \dots))$
- (b) $((1, 1, 1, 1, 1, \dots), (0, 1, 1, 0, a, \dots))$
- (c) $((a, 1, a, 1, a, \dots), (a, 1, a, 0, a, \dots))$
- (d) $((1, 1, 1, 1, 1, \dots), (0, 0, a, a, a, \dots))$
- (e) $((1, 1, 1, 1, 1, \dots), (0, a, 0, a, a, \dots))$

Solution:

- (a) no
- (b) yes
- (c) no
- (d) yes
- (e) no

4.2 E Define the composite state machine in figure 4.9 in terms of the component machines, as done for the simpler compositions in figures 4.2 and 4.1. Be sure to state any required assumptions. Give the definition in two different ways:

- (a) Directly form a product of the three state spaces.
- (b) First compose the A and B state machines to get a new D state machine, and then compose D with C .
- (c) Comment on the relationship between the models in part (a) and (b).

Solution:

(a) **Assumptions about the component machines:**

$$\text{Outputs}_A \subset \text{Inputs}_B \subset \text{Inputs}_C$$

Definition of the composite machine:

$$\text{States} = \text{States}_A \times \text{States}_B \times \text{States}_C$$

$$\text{Inputs} = \text{Inputs}_A$$

$$\text{Outputs} = \text{Outputs}_C$$

$$\text{initialState} = (\text{initialState}_A, \text{initialState}_B, \text{initialState}_C)$$

$$\text{update}((s_A, s_B, s_C), x) = ((s'_A, s'_B, s'_C), y_C)$$

where

$$(s'_A, y_A) = \text{update}_A(s_A, x), (s'_B, y_B) = \text{update}_B(s_B, y_A), \text{ and} \\ (s'_C, y_C) = \text{update}_C(s_C, y_B).$$

(b) For the first composition (A and B):

Assumptions about the component machines:

$$\text{Outputs}_A \subset \text{Inputs}_B$$

Definition of the composite machine:

$$\text{States}_D = \text{States}_A \times \text{States}_B$$

$$\text{Inputs}_D = \text{Inputs}_A$$

$$\text{Outputs}_D = \text{Outputs}_B$$

$$\text{initialState}_D = (\text{initialState}_A, \text{initialState}_B)$$

$$\text{update}_D((s_A, s_B), x) = ((s'_A, s'_B), y_B)$$

where

$$(s'_A, y_A) = \text{update}_A(s_A, x) \text{ and}$$

$$(s'_B, y_B) = \text{update}_B(s_B, y_A).$$

For the second composition (D and C):

Assumptions about the component machines:

$$\text{Outputs}_D \subset \text{Inputs}_C$$

Definition of the composite machine:

$$\text{States} = \text{States}_D \times \text{States}_C$$

$$\text{Inputs} = \text{Inputs}_D$$

$$\text{Outputs} = \text{Outputs}_C$$

$$\text{initialState} = (\text{initialState}_D, \text{initialState}_C)$$

$$\text{update}((s_D, s_C), x) = ((s'_D, s'_C), y_C)$$

where

$$(s'_D, y_D) = \text{update}_D(s_D, x) \text{ and } (s'_C, y_C) = \text{update}_C(s_C, y_D).$$

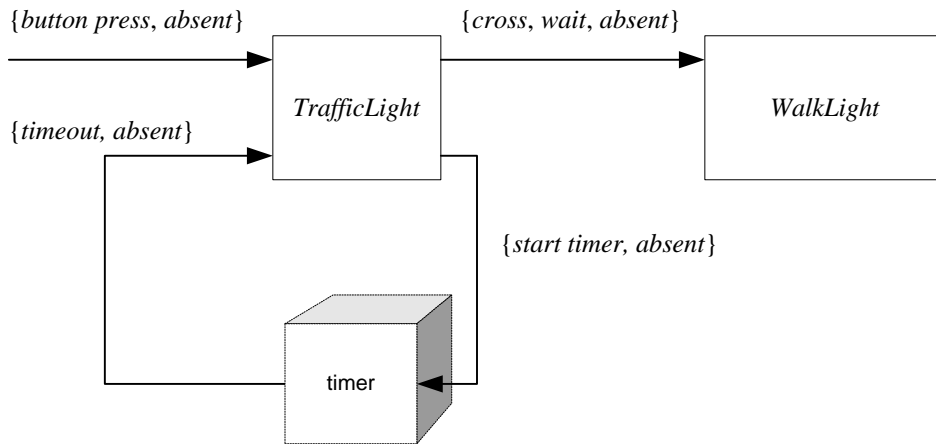
(c) The models in (a) and (b) are different but equivalent state machines.

4.5 C A road has a pedestrian crossing with a traffic light. The light is normally green for vehicles, and the pedestrian is told to wait. However, if a pedestrian presses a button, the light turns yellow for 30 seconds and then red for 30 seconds. When it is red, the pedestrian is told “cross now.” After the 30 seconds of red, the light turns back to green. If a pedestrian presses the button again while the light is red, then the red is extended to a full minute.

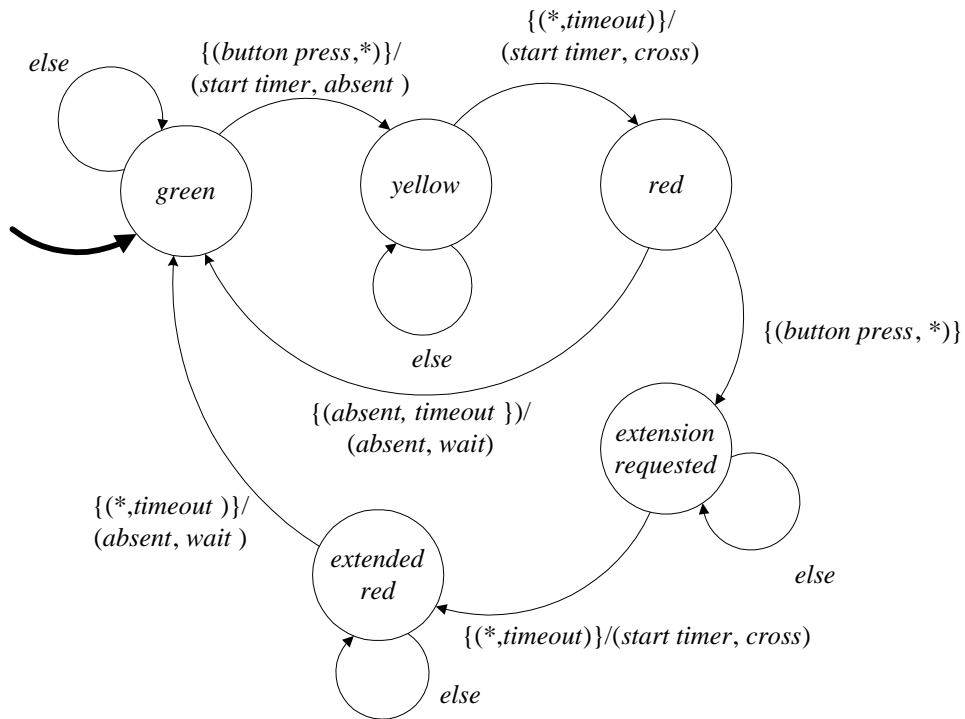
Construct a composite model for this system that has at least two state machines, *TrafficLight* for the traffic light seen by the cars, and *WalkLight* for the walk light seen by the pedestrians. The state of machine should represent the state of the lights. For example, *TrafficLight* should have at least three states, one for green, one for yellow, and one for red. Each color may, however, have more than one state associated with it. For example, there may be more than one state in which the light is red. It is typical in modeling systems for the states of the model to represent states of the physical system.

Assume you have a timer available such that if you emit an output *start timer*, then 30 seconds later an input *timeout* will appear. It is sufficient to give the state transition graphs for the machines. State any assumptions you need to make.

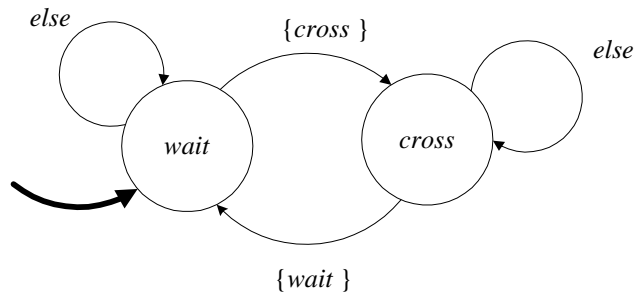
Solution: One possible design structures the state machines as follows:



where the state transition diagram for the traffic light model is:



and the state transition diagram for the wait light is:



This assumes that the crossing light starts in the “wait” state and that outputs from this state machine change it.

The set of states for the traffic light is

$$States = \{green, yellow, red, extension\ requested, extended\ red\},$$

The set of inputs is

$$Inputs = \{button\ press, absent\} \times \{timeout, absent\}.$$

The set of outputs is

$$Outputs = \{start\ timer, absent\} \times \{wait, cross, absent\}.$$

The *WaitLight* machine is more trivial:

$$States = \{wait, cross\},$$

$$Inputs = \{wait, cross, absent\}.$$

There are no outputs, so

$$Outputs = \{absent\}.$$