SBf12, Assignment 6 Solutions

Kevin Oishi

November 13, 2012

1 Problem 1

1.1 Part (a)

$$\emptyset \underset{\gamma}{\overset{\alpha}{\rightleftharpoons}} A \tag{1}$$

$$\emptyset \underset{\gamma}{\overset{I}{\rightleftharpoons}} I \tag{2}$$

$$A + A \underset{k_d}{\overset{k_a}{\rightleftharpoons}} D \tag{3}$$

$$A + I \underset{k_d}{\overset{\eta k_a}{\rightleftharpoons}} H \tag{4}$$

$$D + g_{Off} \underset{k_d}{\overset{k_a}{\rightleftharpoons}} g_{On} \tag{5}$$

$$g_{On} \xrightarrow{\alpha} g_{On} + G \tag{6}$$

$$D \xrightarrow{\gamma} \emptyset \tag{7}$$

$$H \xrightarrow{\gamma} \emptyset \tag{8}$$

$$G \xrightarrow{\gamma} \emptyset \tag{9}$$

1.2 Part (b)

$$\dot{A} = 1 - \gamma - \gamma A - 400\gamma A^2 + 200\gamma D + 100\gamma H - 200\gamma AI \tag{10}$$

$$\dot{D} = 200\gamma A^2 - 101\gamma D - 200\gamma D g_{Off} + 100\gamma g_{On} \tag{11}$$

$$\dot{G} = -\gamma G + 10\gamma g_{On} \tag{12}$$

$$\dot{g}_{Off} = -200\gamma D g_{Off} + 100\gamma g_{On} \tag{13}$$

$$\dot{g}_{On} = 200\gamma D g_{Off} - 100\gamma g_{On} \tag{14}$$

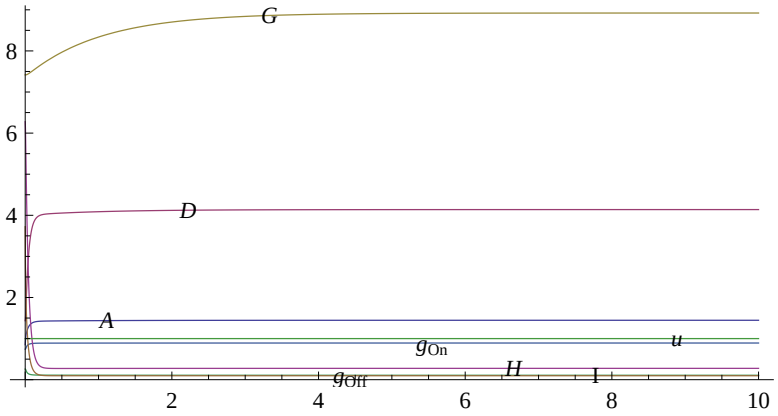
$$\dot{H} = -101\gamma H + 200\gamma AI \tag{15}$$

$$\dot{I} = 10\gamma + 100\gamma H - \gamma I - 200\gamma AI - 100\gamma Iu \tag{16}$$

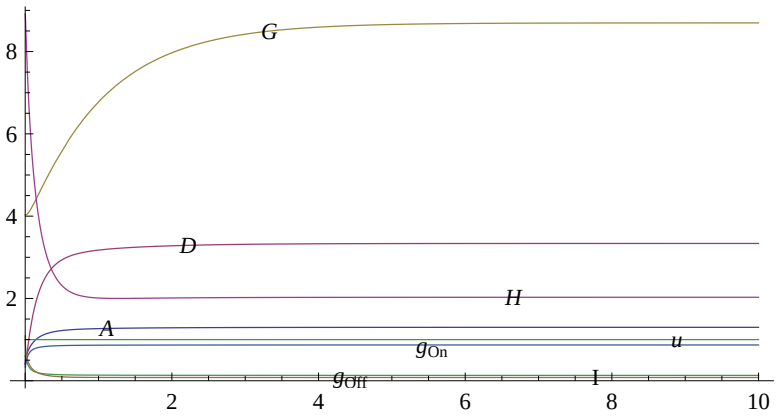
1.3 Part (c)

For this section I simulated the system for $\eta = 1, 10, 100, 1000$, $\gamma = 1$, and $u = 1$. Each trajectory begins at the steady state for $u = 0$. Note that as η increases, the steady state of G decreases.

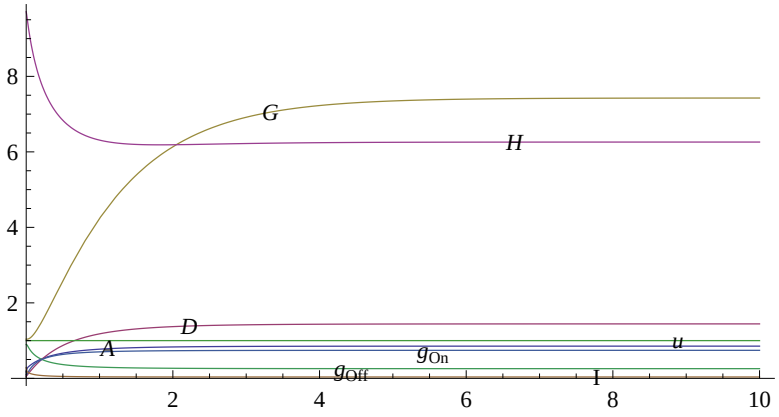
For $\eta = 1$,



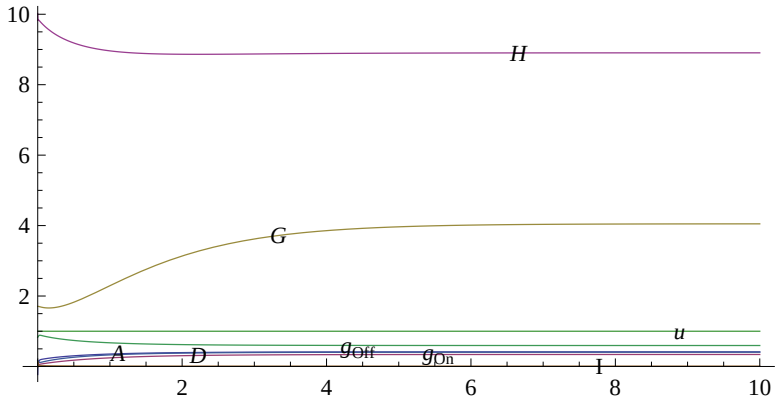
For $\eta = 10$,



For $\eta = 100$,



For $\eta = 1000$,



2 Problem 2

2.1 Part (a)

Using the extended generator,

$$\begin{aligned}
\dot{\mu}_x &= k_1\mu_y - k_2\mu_x && (17)\\
\dot{\mu}_y &= k_1\mu_x - k_2\mu_y && (18)\\
\dot{\mu}_{xx} &= k_1(\mu_y + 2\mu_{xy}) + k_2(\mu_x - 2\mu_{xx}) && (19)\\
\dot{\mu}_{yy} &= k_2(\mu_x + 2\mu_{xy}) + k_2(\mu_y - 2\mu_{yy}) && (20)\\
\dot{\mu}_{xy} &= -2k_2\mu_{xy} + k_2(\mu_{xx} + \mu_{yy}) && (21)
\end{aligned}$$

$$\begin{aligned}
\left\langle (x-\mu_x)^2 \right\rangle &= \left\langle x^2 - 2x\mu_x + \mu^2 \right\rangle && (22)\\
&= \mu_{xx} - \mu_x^2 && (23)\\
\left\langle (y-\mu_y)^2 \right\rangle &= \mu_{yy} - \mu_y^2. && (24)
\end{aligned}$$

$$\frac{d}{dt}\left\langle (x-\mu_x)^2 \right\rangle = \dot{\mu}_{xx} - 2\mu_x\dot{\mu}_x \tag{25}$$

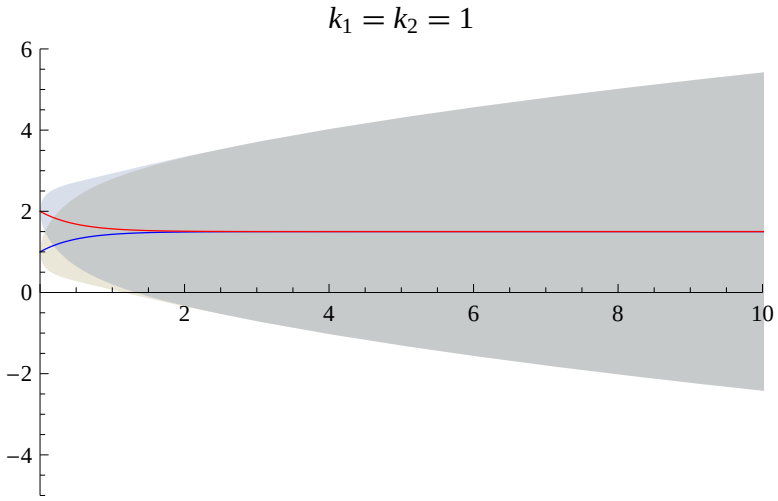
$$= k_2(\mu_x + 2\mu_x^2 - 2\mu_{xx}) + k_1(2\mu_{xy} + \mu_y - 2\mu_x\mu_y) \tag{26}$$

$$\frac{d}{dt}\left\langle (y-\mu_y)^2 \right\rangle = k_2(\mu_y + 2\mu_y^2 - 2\mu_{yy}) + k_1(2\mu_{xy} + \mu_x - 2\mu_x\mu_y) \tag{27}$$

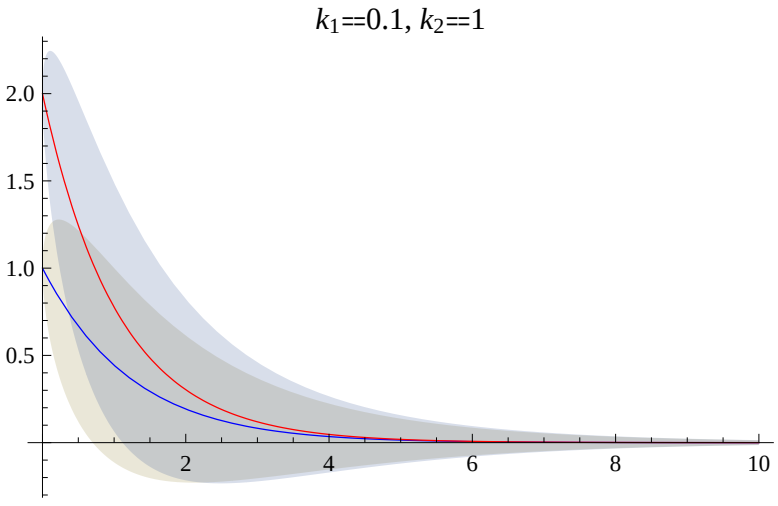
$$\tag{28}$$

2.2 Part (b)

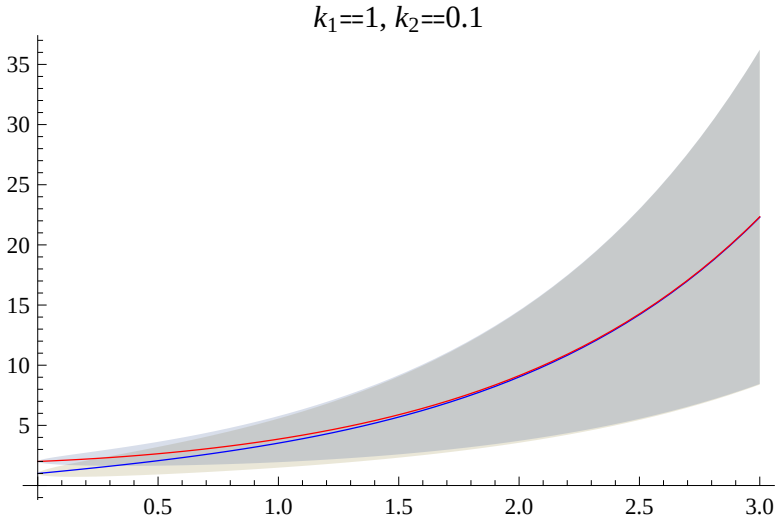
For $k_1 = k_2$ and $x_0 = 1, y_0 = 2$, the mean quickly approaches a steady state where $x = y$ and the variance increases, as illustrated below.



For $k_1 < k_2$ and $x_0 = 1, y_0 = 2$, the mean quickly approaches a steady state where $x = y = 0$ and the variance approaches 0.

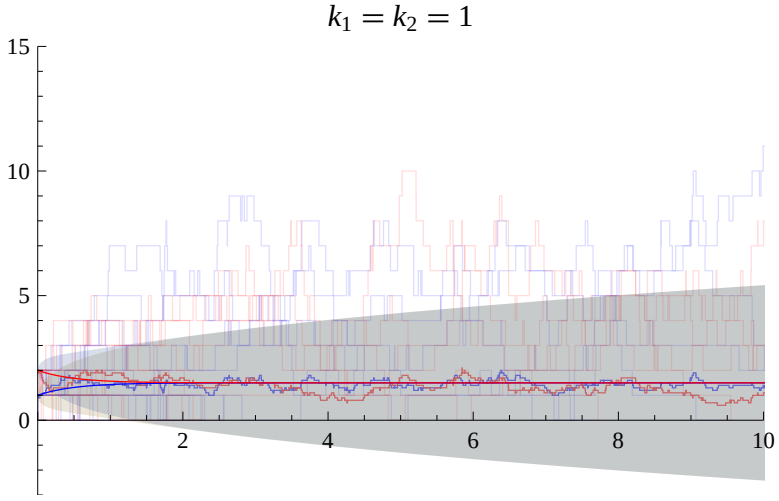


For $k_1 > k_2$ and $x_0 = 1, y_0 = 2$, the both the mean and variance grow as time increases.



2.3 Part (c) [EXTRA CREDIT]

Below are a 15 simulated trajectories, the mean of those trajectories, as well as the first moments and standard deviation windows for $k_1 = k_2$ and $x_0 = 1, y_0 = 2$.



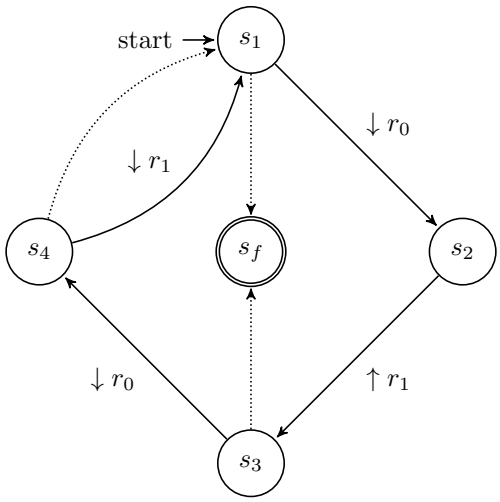
3 Problem 3

3.1 Part (a)

Here is one example of a register machine that computes $mod(r_1, 2)$,

$$dec[s_1, r_0, s_2, s_f] \tag{29}$$
$$inc[s_2, r_1, s_3] \tag{30}$$
$$dec[s_3, r_0, s_4, s_f] \tag{31}$$
$$dec[s_4, r_1, s_1, s_1]. \tag{32}$$

Below is a state transition diagram of the same register machine. Note that I chose the dotted transition from state s_4 to go to s_0. This guarantees that the accepting state s_f will be reached in all trajectories, however another choice would be for the dotted transition to go back to s_4, which guarantees that if register r_1 can be decremented, it will be eventually.



3.2 Part (b)

Below are chemmical reactions that implement this register machine for a clock length of 3 ($l = 3$),

$$s_1 + r_0 \xrightarrow{1} s_2 \tag{33}$$

$$c_1 + s_1 \xrightarrow{1} s_f + c_3 \tag{34}$$

$$c + s_2 \xrightarrow{1} s_3 + r_1 + c \tag{35}$$

$$s_3 + r_0 \xrightarrow{1} s_4 \tag{36}$$

$$c_1 + s_3 \xrightarrow{1} s_f + c_3 \tag{37}$$

$$s_4 + r_1 \xrightarrow{1} s_1 \tag{38}$$

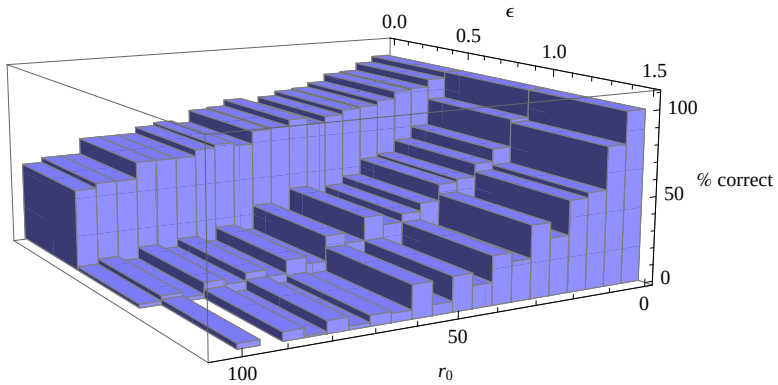
$$c_1 + s_4 \xrightarrow{1} s_1 + c_3 \tag{39}$$

$$c_3 \underset{\epsilon^{-1/2}}{\overset{1}{\rightleftharpoons}} c_2 \tag{40}$$

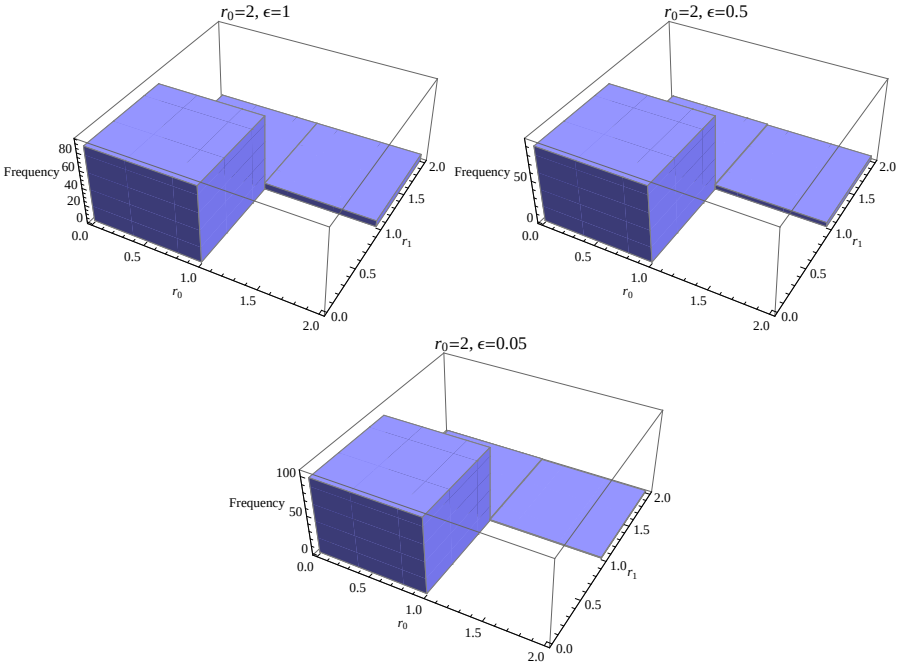
$$c_2 \underset{\epsilon^{-1/2}}{\overset{1}{\rightleftharpoons}} c_1. \tag{41}$$

3.3 Part (c)

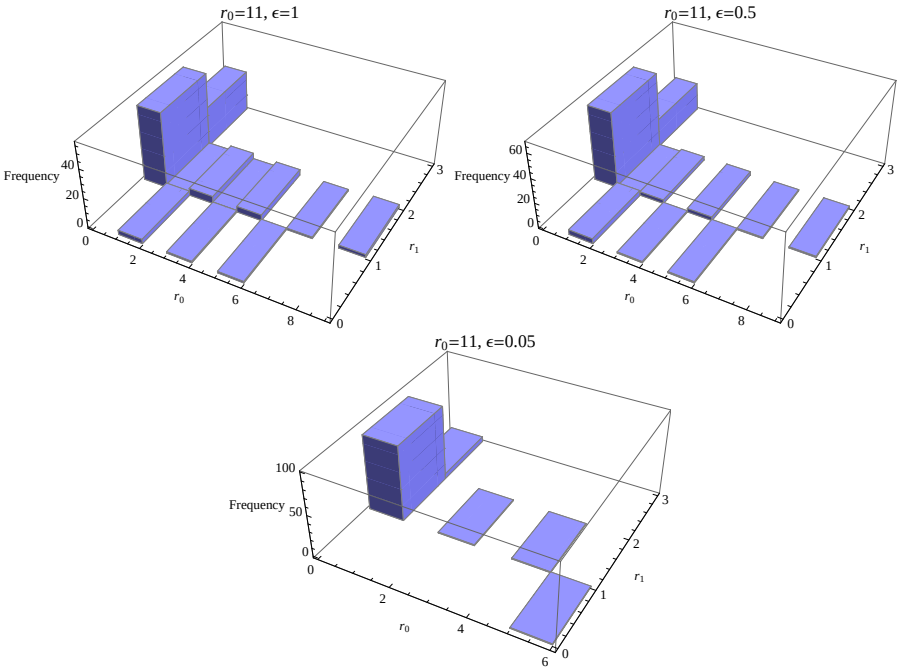
I simulated the above system in gro for $\epsilon = 1, 0.5, 0.05$ with $l = 5$. Below is a histogram showing the percent of correct final computations for $r_0 = 0, 5, 10, \ldots, 100$.



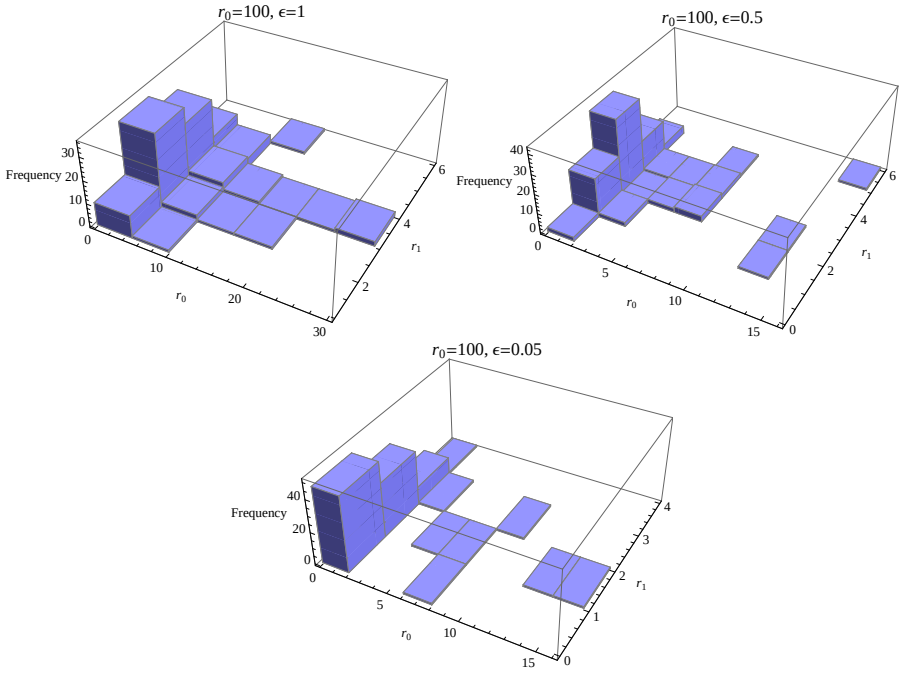
Below are histograms of the register at the accepting state s_f for 100 cells and $r_0 = 2, 11, 100$.



For an initial $r_0 = 2$, a correct evaluation leaves the register in the state $(r_0, r_1) = (0, 0)$. For $\epsilon = 1, 0.5, 0.05$ most trajectories ended with a correct evaluation.



For an initial $r_0 = 11$, a correct evaluation leaves the register in the state $(r_0, r_1) = (1, 0)$. For $\epsilon = 1, 0.5, 0.05$ most trajectories ended with a correct evaluation.



For an initial $r_0 = 100$, a correct evaluation leaves the register in the state $(r_0, r_1) = (0, 0)$. For $\epsilon = 1, 0.5$ most trajectories ended with an incorrect evaluation. For $\epsilon = 0.05$, most trajectories ended with a correct evaluation. Below is the gro code I used to generate this data.

```
include gro

fun arg i default . if (ARGC>i) then atof(ARGV[i]) else default end;

r0 := arg 2 2;
epsilon := arg 3 0.05;
outfile := fopen("/tmp/a6_"<>tostring(r0)<>"_"<>tostring(epsilon)<>".csv","w");

program inc(i,n,j) := {
  needs r,s,c;
  rate(1*s[i]) : {
    s[i] := s[i]-1,
    r[n] := r[n]+1,
    s[j] := s[j]+1
  }
};
```

```
program dec(i,n,j,k) := {
  needs r,s,c;
  rate(s[i]*r[n]) : {
    s[i] := s[i]-1,
    r[n] := r[n]-1,
    s[j] := s[j]+1
  }
  rate(c[0]*s[i]) : {
    s[i] := s[i]-1,
    c[0] := c[0]-1,
    s[k] := s[k]+1,
    c[length(c)-1] := c[length(c)-1]+1
  }
};

program clock(epsilon) := {
  c := {0,0,0,0,1};
  l := length(c);
  rate(c[4]) :                            {c[4] := c[4]-1, c[3] := c[3]+1}
  rate(c[3]*1.0/epsilon^(1.0/(l-1))) :  {c[4] := c[4]+1, c[3] := c[3]-1}
  rate(c[3]) :                            {c[3] := c[3]-1, c[2] := c[2]+1}
  rate(c[2]*1.0/epsilon^(1.0/(l-1))) :  {c[3] := c[3]+1, c[2] := c[2]-1}
  rate(c[2]) :                            {c[2] := c[2]-1, c[1] := c[1]+1}
  rate(c[1]*1.0/epsilon^(1.0/(l-1))) :  {c[2] := c[2]+1, c[1] := c[1]-1}
  rate(c[1]) :                            {c[1] := c[1]-1, c[0] := c[0]+1}
  rate(c[0]*1.0/epsilon^(1.0/(l-1))) :  {c[1] := c[1]+1, c[0] := c[0]-1}
};

program mod2init(x) := {
  needs c;
  r := {x,0};
  s := {1,0,0,0,0};
  //s[4]>0 : {stop()}
};

program reporter() := {
  needs r,s;
  rfp := 0;
  gfp := 0;
  yfp := 0;
  s[4]>0 & r[1]=0 : {gfp := 100*volume}
  s[4]>0 & r[1]=1 : {yfp := 100*volume}
  s[4]>0 & r[1]>1 : {rfp := 100*volume}
  //p := [t:=0];
  //true : {print(id,"\t",s,"\t",r,"\t",p.t,"\n"), p.t := p.t+dt}
```

```
};

program mod2(x,epsilon) := mod2init(x) + clock(epsilon) sharing r,s,c,rfp,gfp,yfp
  + dec(0,0,1,4)  sharing r,s,c,rfp,gfp,yfp
  + inc(1,1,2)    sharing r,s,c,rfp,gfp,yfp
  + dec(2,0,3,4)  sharing r,s,c,rfp,gfp,yfp
  + dec(3,1,0,0)  sharing r,s,c,rfp,gfp,yfp
  + reporter()    sharing r,s,c,rfp,gfp,yfp;

program main() := {
  (sumlist maptocells s[4] end) = 100 : {
    maptocells fprint(outfile,r[0],",",r[1],"\n") end,
    stop()
  }
};

set("ecoli_growth_rate",0);
foreach q in cross (range 10) (range 10) do
  ecoli([x:=q[0]*25-125, y:=q[1]*25-125],program mod2(r0,epsilon))
end;
```