用Matlab实现元胞自动机（网上收集、转载）技术 2010-03-13 13:04:28 阅读121 评论0 字号：大中小 订阅 .

file:life.m

%% 初始化

m = 50;

X = zeros(m,m);

X(25,25) = 1;

n = [m 1:m-1];

e = [2:m 1];

s = [2:m 1];

w = [m 1:m-1];

% 绘制初始图形

[i,j] = find(X);

figure(gcf);

plothandle = plot(i,j,'.', ...

'Color','blue', ...

'MarkerSize',12);

axis([0 m+1 0 m+1]);

%% 演化

for k = 1:50

%邻居数

N = X(n,:) + X(s,:) + X(:,e) + X(:,w) + ...

X(n,e) + X(n,w) + X(s,e) + X(s,w);

%概率阵

RAND = rand(m);

%换代

X = X | (N.\*RAND>0.99);

%绘图

[i,j] = find(X);

set(plothandle,'xdata',i,'ydata',j)

drawnow

pause(0.2)

k

end

file 2:

function sierpinski(n);

% 使用元胞自动机生成sierpinski直角垫片

% Example:

% sierpinski(256);

% %算法见:孙博文,《分形算法与程序设计：用Visual C++实现》

if nargin==0;

n=256;

end

X=ones(n);

X(1,n-1)=0;

H=imshow(X,[]);

set(gcf,'doublebuffer','on');

k=1;

while k<n;

X(k+1,1:end-1)=xor(X(k,1:end-1),X(k,2:end));

X(k+1,n)=1;

set(H,'CData',X);

pause(0.1);

k=k+1;

end

file 3:

function CA\_sim\_cloud;

% 使用元胞自动机模拟地球卫星的云图

%

% reference:

% Piazza, E.; Cuccoli, F.;

% Cellular Automata Simulation of Clouds in Satellite Images,

% Geoscience and Remote Sensing Symposium, 2001. IGARSS '01.

% IEEE 2001 International Volume 4, 9-13 July 2001 Page(s):

% 1722 - 1724 vol.4 Digital Object Identifier 10.1109/IGARSS.

% 2001.977050

time=888; % 程序执行步数

M=240;

N=320;

S=round(rand(M,N)\*15);

p=[1,2,1,6,6,1,2,1];

p=sum(tril(meshgrid(p)),2)/20;

rand('state',0);

SS=S;

R=rand(M,N);

G=R;

B=R;

C=cat(3,R,G,B);

fig=figure;

set(fig,'DoubleBuffer','on');

mov = avifile('example2.avi');

cc=imshow(C,[]);

set(gcf,'Position',[13 355 157 194])

x1=(1:3)+round(M/2);y1=(1:3)+round(N/3);

x2=(1:3)+round(M/3);y2=(1:3)+round(N/2);

x3=(1:3)+round(M/1.5);y3=(1:3)+round(N/2);

q=0;

qq=15/4;

while q<time;

SS=zeros(M,N);

for k=1:15;

r=rand(M,N); % 生成几率r

K=zeros(M+2,N+2);

T=(S-k>=0); % 粒子数矩阵

K(2:end-1,2:end-1)=T;

SS=K(1:end-2,1:end-2).\*(r<p(1))+...

K(1:end-2,2:end-1).\*(r<p(2) & r>=p(1))+...

K(1:end-2,3:end).\*(r<p(3) & r>=p(2))+...

K(2:end-1,1:end-2).\*(r<p(4) & r>=p(3))+...

K(2:end-1,3:end).\*(r<p(5) & r>=p(4))+...

K(3:end,1:end-2).\*(r<p(6) & r>=p(5))+...

K(3:end,2:end-1).\*(r<p(7) & r>=p(6))+...

K(3:end,3:end).\*(r>=p(7))+SS;

end

S=SS; %SS是粒子扩散后的分布

S(S>15)=15;

S(x1,y1)=15;

S(x2,y2)=15;

S(x3,y3)=15; % 粒子源赋值

G=(S<=7.5);

B=(S>qq);

R=(S>qq & S<=7.5);

C=double(cat(3,R,G,B));

set(cc,'CData',C);

q=q+1;

pause(0.2);

title(['q=',num2str(q)]);

Nu(q)=sum(S(1:end));

F = getframe(gca);

mov = addframe(mov,F);

end

mov = close(mov);

figure;

plot(Nu)

file 4:

题目: 六边形的元胞自动机上的单粒子运动

摘要: 本程序在六边形的元胞自动机上模拟单粒子运动,算法是基于FHP规则.元胞自动机模拟地球卫星的云图

关键词: 六边形, 元胞自动机, FHP规则

figure('Position',[15 30 997 658],'NumberTitle','off');

set(gcf,'name',' 六边形的元胞自动机上的单粒子运动');

% Author's email: zjliu2001@163.com

% Reference:

% U. Frisch, B. Hasslacher, Y. Pomeau, Lattice-gas

% automata for the Navier-Stokes rquation, Phys. Rev.

% Lett. 1986,56: 1505-1508

set(gcf,'DoubleBuffer','on');

axis square;box on;

set(gca,'XColor','r','YColor','r');

set(gca,'Position',[-0.01 0.11 0.775 0.815]);

L=17.5\*0.1/sqrt(3);

axis([0,L,0,1]); hold on;

for p=0:.1:0.9;

plot([0,(1-p)/sqrt(3)],[p,1],'k');

end

for p=0:0.1/sqrt(3):1;

plot([p,min(p+1/sqrt(3),17.5\*0.1/sqrt(3))],[0,min(1,(L-p)\*sqrt(3))],'k');

end

for p=0:0.1/sqrt(3):1;

plot([0,p],[p\*sqrt(3),0],'k');

end

for p=0:9;

plot([L-[0.05+p/10]/sqrt(3),L],[1,1-[0.05+p/10]],'k');

end

for p=0:0.05:1;

plot([0,L],[p,p],'k');

end

po=plot(0.8/sqrt(3),0.5,'r.','markersize',24);

pz=0.8/sqrt(3)+0.5i; % the position of read point

A=pi/3\*2; % the movement direction of read point

gc=gca;

a1=axes('Position',[0.7,0.5,0.25,0.3]);

axis square;hold on;axis([0,1,0,1]);

plot([0.5+0.5i,(1+i)/2+0.4\*exp(i\*pi/3\*2)]);

plot([0.5+0.5i,(1+i)/2+0.4\*exp(i\*pi/3)]);

plot([0.3,0.7],[0.5,0.5]);

text(0.2,0.8,'Y','fontsize',14);

text(0.73,0.8,'X','fontsize',14);

text(0.2,0.4,'Z','fontsize',14);

axes(gc);

dt=0.1/sqrt(3); k=0;

ses=['while k;',...

'pz=pz+dt\*exp(i\*A);',...

'if imag(pz)>0.99 | imag(pz)<0.01;',...

' A=-A;',...

'end;',...

'if real(pz)>0.99 | real(pz)<0.01;',...

' A=-A-pi;',...

'end;',...

'set(po,''XData'',real(pz),''YData'',imag(pz));',...

'pause(0.2);',...

'end;'];

po1=uicontrol(gcf,'style','push',...

'unit','normalized','position',[0.74,0.87,0.1,0.08],...

'string','start','fontsize',18,'callback',[]);

set(po1,'callback',['k=~k;if k==1;',...

'set(po1,''string'',''stop'');',...

'else set(po1,''string'',''start'');',...

'end;',ses]);

file 5:

% DLA

%%%%%来源：萝卜驿站 http://luobo.ycool.com/

clc;clear;close all;

S=ones(40,100);% state matrix

S(end,:)=0; % initial sttae

Ss=zeros(size(S)+[1,0]); % top line is origin of particle

Ss(2:end,:)=S; % showing matrix

N=size(S,2);

II=imagesc(Ss);axis equal;colormap(gray)

set(gcf,'DoubleBuffer','on');

while sum(1-S(1,:))<0.5;

y=1;

x=round(rand\*[N-1]+1); % random position

D=0;

while D<0.5; % random travel

r=rand;

if abs(x-1)<0.1;

SL=1;

else

SL=S(y,x-1);

end

if abs(x-N)<0.1;

SR=1;

else

SR=S(y,x+1);

end

if SL+SR+S(y+1,x)<2.5; % check the neighbor: left, right, under

D=1;

S(y,x)=0; % stop in the position

end

if r<=1/3; % travel randomly

x=x-1;

elseif r<=2/3;

x=x+1;

else

y=y+1;

end

Ss(2:end,:)=S;

if x<0.5|x>N+0.5;

D=1; % out of the range

else

Ss(y,x)=0; % to show the moving particle

end

set(II,'CData',Ss); % to show

pause(0.1);

end

end

file 6:

function sands(N);

% 砂堆规则

% 参考书目：

% 物理系统的元胞自动机模拟

% 作者:(英国)肖帕德等著、祝玉学等译

close all

figure;

set(gcf,'Doublebuffer','on');

D=ones(N);

D1=D;

[X,Y]=meshgrid(1:N);

Z=2\*X-Y;

p=fix(9.5\*N/21);

D(Z>p & Z<9+p)=0;

D(fix(end/2)+1:end,:)=1;

D=min(D,flipud(D).\*D1);

D=min(D,fliplr(D).\*D1);

D(end,fix(end/4.13):end-fix(end/4.13))=0;

D(end-1,fix(end/4.12):end-fix(end/4.12))=0;

D(end-2,fix(end/4.1):end-fix(end/4.1))=0;

% imshow(D,[])

% 以上是生成装砂的瓶子

B=ones(N);

B(Z>9+p)=0;

B(fix(end/2)+1:end,:)=1;

B(:,fix(end/2)+1:end)=1;

B=min(B,fliplr(B));

B(1:3,:)=1;

% figure;

set(gcf,'Doublebuffer','on');

% imshow(B,[])

% mov = avifile('example2900.avi');

kk=3800;

for k=1:kk;

[B,Nu]=duisha(D,B);

Bk=min(D,B);

imshow(Bk,[])

Ha(k)=Nu;

% F = getframe(gca);

% mov = addframe(mov,F);

end

% mov = close(mov);

figure;

plot(Ha)

function [Y,Nu]=duisha(D,B);

Dq=10\*(1-D);

Bg=1-B+Dq;

% 研究砂子下落

S=zeros(size(B));

S1=S;

S2=S;

S(2:end,:)=Bg(2:end,:)-Bg(1:end-1,:);

S1(S==-1)=1;

S2(1:end-1,:)=S1(2:end,:);

Bg(S1==1)=~Bg(S1==1);

Bg(S2==1)=~Bg(S2==1);

% 研究砂子倾倒

clear S

clear S1

clear S2

S=zeros(size(B));

S1=S;

S2=S;

S(1:end-1,2:end-1)=Bg(1:end-1,2:end-1)+Bg(2:end,2:end-1)-Bg(2:end,1:end-2);

S1(S==2)=1;

S2(2:end,1:end-2)=S1(1:end-1,2:end-1);

Bg(S1==1)=0;

Bg(S2==1)=1;

clear S

clear S1

clear S2

S=zeros(size(B));

S1=S;

S2=S;

S(1:end-1,2:end-1)=Bg(1:end-1,2:end-1)+Bg(2:end,2:end-1)-Bg(2:end,3:end);

S1(S==2)=1;

S2(2:end,3:end)=S1(1:end-1,2:end-1);

Bg(S1==1)=0;

Bg(S2==1)=1;

Y=(1-Bg).\*D;

Nu=prod(size(find(Y==0)));

file 7:

function CA\_sim\_cloud;

% 使用元胞自动机模拟地球卫星的云图

%

% reference:

% Piazza, E.; Cuccoli, F.;

% Cellular Automata Simulation of Clouds in Satellite Images,

% Geoscience and Remote Sensing Symposium, 2001. IGARSS '01.

% IEEE 2001 International Volume 4, 9-13 July 2001 Page(s):

% 1722 - 1724 vol.4 Digital Object Identifier 10.1109/IGARSS.

% 2001.977050

time=500; % 程序执行步数

M=240;

N=320;

S=zeros(M,N);

p=[1,2,1,6,6,1,2,1];

p=sum(tril(meshgrid(p)),2)/20;

rand('state',0);

SS=S;

R=1-S;

G=S;

B=S;

C=cat(3,R,G,B);

figure;

cc=imshow(C,[]);

x=round(M/2);y=(1:3)+round(N/3);

q=0;

while q<time;

SS=zeros(M,N);

for k=1:15;

r=rand(M,N); % 生成几率r

K=zeros(M+2,N+2);

T=(S-k>=0); % 粒子数矩阵

K(2:end-1,2:end-1)=T;

SS=K(1:end-2,1:end-2).\*(r<p(1))+...

K(1:end-2,2:end-1).\*(r<p(2) & r>=p(1))+...

K(1:end-2,3:end).\*(r<p(3) & r>=p(2))+...

K(2:end-1,1:end-2).\*(r<p(4) & r>=p(3))+...

K(2:end-1,3:end).\*(r<p(5) & r>=p(4))+...

K(3:end,1:end-2).\*(r<p(6) & r>=p(5))+...

K(3:end,2:end-1).\*(r<p(7) & r>=p(6))+...

K(3:end,3:end).\*(r>=p(7))+SS;

end

S=SS; %SS是粒子扩散后的分布

S(S>15)=15;

S(x,y)=15; % 粒子源赋值

G=(S<=10);

B=(S>5);

R=(S>5 & S<=10);

C=double(cat(3,R,G,B));

set(cc,'CData',C);

q=q+1;

pause(0.2);

title(['q=',num2str(q)]);

Nu(q)=sum(S(1:end));

end

figure;

plot(Nu)

file 8:

生命游戏 (Came of Life)是J. H. Conway在20世纪60年代末设计的一种单人玩的计算机游戏(Garclner，M.，1970、1971)。他与现代的围棋游戏在某些特征上略有相似：围棋中有黑白两种棋子。生命游戏中的元胞有{"生"，"死"}两个状态 {0，1};围棋的棋盘是规则划分的网格，黑白两子在空间的分布决定双方的死活，而生命游戏也是规则划分的网格(元胞似国际象棋分布在网格内。而不象围棋的棋子分布在格网交叉点上)。根据元胞的局部空间构形来决定生死。只不过规则更为简单。下面介绍生命游戏的构成及规则:

(1)元胞分布在规则划分的网格上；

(2)元胞具有0，1两种状态，0代表"死"，l代表"生"；

(3)元胞以相邻的8个元胞为邻居。即Moore邻居形式；

(4)一个元胞的生死由其在该时刻本身的生死状态和周围八个邻居的状态 (确切讲是状态的和)决定:

·在当前时刻，如果一个元胞状态为"生"，且八个相邻元胞中有两个或三个的状态为"生"，则在下--时刻该元胞继续保持为"生"，否则"死"去；

·在当前时刻。如果一个元胞状态为"死"。且八个相邻元胞中正好有三个为"生"。则该元胞在下一时刻 "复活"。否则保持为"死"。

尽管它的规则看上去很简单。但生命游戏是具有产生动态图案和动态结构能力的元胞自动机模型。它能产生丰富的、有趣的图案。生命游戏的优化与初始元胞状态值的分布有关，给定任意的初始状态分布。经过若干步的运算，有的图案会很快消失。而有的图案则固定不动，有的周而复始重复两个或几个图案，有的婉蜒而行。有的则保持图案定向移动，形似阅兵阵……，其中最为著名的是"滑翔机 (叫Glider)"的图案。

matlab程序如下：

% 其中黑点表示活着

% 白点表示死亡状态

function Game\_of\_Life(n)

% 生命游戏

% Example:

% Game\_of\_Life(100);

if nargin==0;

n=100;

end

B=round(rand(n+2));

Z=B(2:end-1,2:end-1);

H=imshow(Z,[]);

set(gcf,'position',[241 132 560 420])

set(gcf,'doublebuffer','on');

xlabel('Please press "space" key and stop this program!',...

'fontsize',12,'color','r');

k=1;

title('Game of life','color','b');

while k;

s=get(gcf,'currentkey');

if strcmp(s,'space');

clc;k=0;

end

A=sumfun(B);

X=zeros(n);

X(Z==1 & (A==2 | A==3))=1;

X(Z==0 & A==3)=1;

B(2:end-1,2:end-1)=X;

Z=X;

set(H,'CData',1-X);

pause(0.5);

end

figure(gcf);

function S=sumfun(B);

% 周围8个位置的和

S=B(1:end-2,2:end-1)+...

B(3:end,2:end-1)+...

B(2:end-1,1:end-2)+...

B(2:end-1,3:end)+...

B(2:end-1,1:end-2)+...

B(1:end-2,3:end)+...

B(3:end,1:end-2)+...

B(3:end,3:end);

file 9:

如果x([i-1,i,i+1],t)等于[0 0 1]或者[1 0 0]时，x(i,t+1)才等于1，其他情况x(i,t+1)等于0。

也就是:

本行 0 0 1 1 0 0 其它

下一行 1 1 0

function sierpinski3\_by\_CA(n);

% 使用元胞自动机生成sierpinski直角垫片

% Example:

% sierpinski3\_by\_CA(256);

% %算法见:孙博文,《分形算法与程序设计：用Visual C++实现》

if nargin==0;

n=256;

end

X=zeros(n);

X(1,round(n/2))=1;

H=imshow(X,[]);

set(gcf,'doublebuffer','on');

k=1;

while k<round(n/2);

X(k+1,2:end-1)=and(xor(X(k,1:end-2),X(k,3:end)),...

~X(k,2:end-1));

set(H,'CData',1-X);

pause(0.05);

k=k+1;

end

nm=round(n/2);

k=1;

while k<nm;

X(nm+k,1:end)=X(nm-k,1:end);

set(H,'CData',1-X);

pause(0.05);

k=k+1;

end

file 10:

function sierpinski(n);

% 使用元胞自动机生成sierpinski直角垫片

% Example:

% sierpinski(256);

% %算法见:孙博文,《分形算法与程序设计：用Visual C++实现》

if nargin==0;

n=256;

end

X=ones(n);

X(1,n-1)=0;

H=imshow(X,[]);

set(gcf,'doublebuffer','on');

k=1;

while k<n;

X(k+1,1:end-1)=xor(X(k,1:end-1),X(k,2:end));

X(k+1,n)=1;

set(H,'CData',X);

pause(0.1);

k=k+1;

end

file 11:

function Clumsychild(c1,m);

% 实现元胞以固定的概率向相邻的4个元胞扩散

% Example:

% Clumsychild(0.1,0.1);

% Author's email:zjliu2001@163.com

N=100;rand('state',0);

A=randperm(N^2);

S=zeros(N);

S(A(1:3000))=3; % 30%的位置是状态3

S(A(3001:5000))=1; % 20%的位置是状态1

S(A(5001:6000))=2; % 10%的位置是状态2

clear A;close all;

figure('position',[159 42 567 427]);

imagesc([1:4]')

set(gca,'YAxisLocation','right');

set(gca,'YAxisLocation','right');

set(gca,'position',[0.8,0.1,0.1,0.8])

set(gca,'position',[0.84,0.12,0.1,0.8])

set(gca,'xtick',[]);

set(gca,'ytick',[1:4]);

set(gca,'yticklabel',num2str([0:3]'));

axes('position',[0.06,0.12,0.7,0.8]);

H=imagesc(S);

set(gcf,'position',[159 42 485 427]);

set(gcf,'doublebuffer','on');

xlabel('Please press "space" key and stop this program!',...

'fontsize',12,'color','r');

title(['c1=',num2str(c1),' m=',num2str(m)]);

k=1;

while k

pause(0.5);

s=get(gcf,'currentkey');

if strcmp(s,'space');

clc;k=0;

end

S=evolvement(S,c1,m);

set(H,'CData',S);

end

figure(gcf);

function S=evolvement(S,c1,m);

P=zeros(size(S));

Da=rand(size(S));

Da(Da>1-c1)=1;

Da(Da<1-c1)=0;

P(S==1 | S==2)=1;

R=round(rand(size(S))+1);

P=P.\*R.\*Da;

V=round(rand(size(S))\*3)+1;

V=V.\*P; % V是速度方向:

% 1 --- up

% 2 --- down

% 3 --- left

% 4 --- right

V(1,V(1,1:end)==1)=0;

V(end,V(end,1:end)==2)=0;

V(V(1:end,1)==3,1)=0;

V(V(1:end,end)==4,end)=0;

% 产生后代

[x,y]=find(V==1);

DD=zeros(size(S));

DD(x-1,y)=P(x-1,y);

S(S==0 | S==2 & DD==1)=1;

S(S==0 & DD==2)=2;

[x,y]=find(V==2);

DD=zeros(size(S));

DD(x+1,y)=P(x+1,y);

S(S==0 | S==2 & DD==1)=1;

S(S==0 & DD==2)=2;

[x,y]=find(V==3);

DD=zeros(size(S));

DD(x,y-1)=P(x,y-1);

S(S==0 | S==2 & DD==1)=1;

S(S==0 & DD==2)=2;

[x,y]=find(V==4);

DD=zeros(size(S));

DD(x,y+1)=P(x,y+1);

S(S==0 | S==2 & DD==1)=1;

S(S==0 & DD==2)=2;

Dr=rand(size(S));

S(S<3 & Dr<m)=0;

file 12:

实现元胞以固定的概率向相邻的4个元胞扩散 zz问题，元胞自动机模拟地球卫星的云图 元胞以固定的概率向相邻的4个元胞扩散 zz问题，元胞自动机模拟地球卫星的云图

萝卜 @ 2005-06-16 08:36

这个完整的元胞自动机模型如下：

定义一个100×100的格子，每个元胞都有4个状态，即0，1，2，和3。其中如果处于0状态的

话，我们认为这个格子是空白，且可以居住的。如果处于1状态，我们认为该点为强物种1所

占有，如果处于2状态，我们认为该点为弱物种2所占有。而状态3表明了是一个被破坏了的地

点，不能被任何物种占有。

1） 如果处于状态1（或者2），那么该点的个体以c1（或者c2）的速度产生后代。

2） 物种i的后代随机地到达上下左右四个元胞，如果这个后代是物种1的，那么他可以扩

散到可以居住的元胞上（为空的元胞或者是被物种2所占有的元胞，即状态为0或者2）如果

这个后代是物种2的，那么他只能扩散到空白且可以居住的元胞中。而如果后代扩散到被毁

坏的元胞（状态为3）的，那么将会失败。

3） 如果一个元胞处于1或者2的状态，这个元胞将以m的概率灭绝。

也就是说，强物种以c1的速度产生后代，并将这个后代随机地发送到相邻的元胞中，强物种

1的可以入侵到空白的元胞中，或者被弱物种2所占的地方。弱物种2以c2的速度产生后代，

并把后代随机的送入其相邻的元胞中，弱物种的后代只能入侵到空白的元胞中（状态为0）。

如果这个后代入侵到毁坏的元胞中（状态为3），那么这个入侵不成功，后代将灭绝。另外，

对于2个物种来说，个体都有一个密度依赖的死亡率。

对于元胞得初始状态，我们定义为：

有D＝0.3的栖息地遭受破坏，也就是30％的格子（元胞）不能为物种所占领，即他们的状态

为3，赋值为3。另外物种1的比例为0.2，即有20％的格子为物种1所占领，状态为1；物种2

的比例为0.1，即有10％的格子为物种2所占领，状态为2；而空白的格子为0.4，即有40％的

格子可供物种居住，但尚未被占领，其状态为0。

file 13:

仿真移动机器人避障的实验

萝卜 @ 2006-02-07 13:32

% 仿真移动机器人避障的实验

close all;clc;

axes('position',[0.1,0.15,0.56,0.7]);

% Author's email: zjliu2001@163.com

set(gcf,'DoubleBuffer','on');

N=50;

A=ones(N,N,3);

xn=round((N+1)/2);

yn=xn;

A(xn,yn,:)=0;

A(10:13,10:13,2:3)=0;

A(40:43,10:13,2:3)=0;

A(10:13,40:43,2:3)=0;

A(40:43,40:43,2:3)=0;

H=imshow(A);

k=0;

p=[1,-1,i,-i];

ss=['while k==1;',...

't=round(3\*rand)+1;x=real(p(t));',...

'y=imag(p(t));A(xn,yn,:)=1;',...

'xn=xn+x;yn=yn+y;',...

'if xn<1 | xn>N;xn=mod(xn,N)+1;end;',...

'if yn<1 | yn>N;yn=mod(xn,N)+1;end;',...

'if A(xn,yn,2)==0;',...

'xn=xn-x;yn=yn-y;A(xn,yn,:)=0;',...

'else A(xn,yn,:)=0;end;',...

'set(H,''CData'',A);',...

'pause(0.2);end;'];

po1=uicontrol(gcf,'style','push',...

'unit','normalized','position',[0.74,0.6,0.2,0.08],...

'string','start','fontsize',12,'callback',[]);

set(po1,'callback',['k=~k;if k==1;',...

'set(po1,''string'',''stop'');',...

'else set(po1,''string'',''start'');',...

'end;',ss]);