
APPENDIX to:

Yuri Tarnopolsky

PATTERN CHEMISTRY OF THOUGHT AND SPEECH AND THEIR
HYPOTHETICAL ANCESTOR

main page:

<http://spirospero.net/thought-and-speech.pdf> and
<http://www.scribd.com/doc/26045810/Yuri-Tarnopolsky-PATTERN-CHEMISTRY-OF-THOUGHT-AND-SPEECH>

Check for updates; **last update: November, 2010**

APPENDIX contains workspace and MATLAB scripts

scripts as text file: <http://spirospero.net/eve-scripts.txt>

scripts as pdf <http://spirospero.net/eve-scripts.pdf>

<http://www.scribd.com/doc/26046555/APPENDIX-to-PATTERN-CHEMISTRY-OF-THOUGHT-AND-SPEECH>

CONTENTS

- Part 1 Workspace
- Part 2 SCALE
- Part 3 EVE
- Part 4 Example of a batch command
- Part 5 Structure of WORLD (62x62)

Part 1. WORKSPACE

RECONSTRUCTION OF WORKSPACE FOR EVE

The workspace for **EVE** needs only **WW.mat** (62 x 62) and **NAMES.mat** (62 x 12)

1. RECONSTRUCTING WW

Connectivity list generated from sparse(WW)

```
WWM= [ 29,2 , 30,2 , 35,2 , 36,2 , 29,3 , 30,4 , 31,4 , 32,4 , 33,4 , 34,4 , 31,5 , 32,6 , 37,6 , 41,6 , 45,6 ,  
49,6 , 52,6 , 56,6 , 62,6 , 33,7 , 34,8 , 35,9 , 36,10 , 37,10 , 38,10 , 39,10 , 38,11 , 39,12 , 40,12 , 40,13 ,  
41,14 , 42,14 , 42,15 , 43,15 , 46,15 , 47,15 , 43,16 , 44,16 , 44,17 , 48,17 , 50,17 , 45,18 , 46,18 ,  
58,18 , 47,19 , 48,19 , 49,19 , 51,20 , 54,20 , 60,20 , 50,21 , 51,21 , 52,21 , 53,22 , 55,22 , 59,22 ,  
61,22 , 53,23 , 54,23 , 55,24 , 56,24 , 57,24 , 57,25 , 58,26 , 59,27 , 60,27 , 61,28 , 62,28 , 2,29 ,  
3,29 , 30,29 , 35,29 , 36,29 , 2,30 , 4,30 , 29,30 , 31,30 , 32,30 , 33,30 , 34,30 , 35,30 , 36,30 ,  
4,31 , 5,31 , 30,31 , 32,31 , 33,31 , 34,31 , 4,32 , 6,32 , 30,32 , 31,32 , 33,32 , 34,32 , 37,32 ,  
41,32 , 45,32 , 49,32 , 52,32 , 56,32 , 62,32 , 4,33 , 7,33 , 30,33 , 31,33 , 32,33 , 34,33 , 4,34 ,  
8,34 , 30,34 , 31,34 , 32,34 , 33,34 , 2,35 , 9,35 , 29,35 , 30,35 , 36,35 , 2,36 , 10,36 , 29,36 ,  
30,36 , 35,36 , 37,36 , 38,36 , 39,36 , 6,37 , 10,37 , 32,37 , 36,37 , 38,37 , 39,37 , 41,37 , 45,37 ,  
49,37 , 52,37 , 56,37 , 62,37 , 10,38 , 11,38 , 36,38 , 37,38 , 39,38 , 10,39 , 12,39 , 36,39 , 37,39 ,  
38,39 , 40,39 , 12,40 , 13,40 , 39,40 , 6,41 , 14,41 , 32,41 , 37,41 , 42,41 , 45,41 , 49,41 , 52,41 ,  
56,41 , 62,41 , 14,42 , 15,42 , 41,42 , 43,42 , 46,42 , 47,42 , 15,43 , 16,43 , 42,43 , 44,43 , 46,43 ,  
47,43 , 16,44 , 17,44 , 43,44 , 48,44 , 50,44 , 6,45 , 18,45 , 32,45 , 37,45 , 41,45 , 46,45 , 49,45 ,  
52,45 , 56,45 , 58,45 , 62,45 , 15,46 , 18,46 , 42,46 , 43,46 , 45,46 , 47,46 , 58,46 , 15,47 , 19,47 ,  
42,47 , 43,47 , 46,47 , 48,47 , 49,47 , 17,48 , 19,48 , 44,48 , 47,48 , 49,48 , 50,48 , 6,49 , 19,49 ,  
32,49 , 37,49 , 41,49 , 45,49 , 47,49 , 48,49 , 52,49 , 56,49 , 62,49 , 17,50 , 21,50 , 44,50 , 48,50 ,  
51,50 , 52,50 , 20,51 , 21,51 , 50,51 , 52,51 , 54,51 , 60,51 , 6,52 , 21,52 , 32,52 , 37,52 , 41,52 ,  
45,52 , 49,52 , 50,52 , 51,52 , 56,52 , 62,52 , 22,53 , 23,53 , 54,53 , 55,53 , 59,53 , 61,53 , 20,54 ,  
23,54 , 51,54 , 53,54 , 60,54 , 22,55 , 24,55 , 53,55 , 56,55 , 57,55 , 59,55 , 61,55 , 6,56 , 24,56 ,  
32,56 , 37,56 , 41,56 , 45,56 , 49,56 , 52,56 , 55,56 , 57,56 , 62,56 , 24,57 , 25,57 , 55,57 , 56,57 ,  
18,58 , 26,58 , 45,58 , 46,58 , 22,59 , 27,59 , 53,59 , 55,59 , 60,59 , 61,59 , 20,60 , 27,60 , 51,60 ,  
54,60 , 59,60 , 22,61 , 28,61 , 53,61 , 55,61 , 59,61 , 62,61 , 6,62 , 28,62 , 32,62 , 37,62 , 41,62 ,  
45,62 , 49,62 , 52,62 , 56,62 , 61,62 ];
```

Reconstruction:

1. enter: WWM

2. enter:

```
WW=zeros(62,62); lwwm=length(WWM);  
for j = (1 : 2 : lwwm), wx = WWM(j); wy = WWM(j+1); WW(wx, wy)=1; end
```

Check with the original:

```
>> sparse(WW)
```

ans =

```
(29,2) 1      (47,19) 1      (4,32) 1      (41,37) 1      (43,44) 1      (62,49) 1  
(30,2) 1      (48,19) 1      (6,32) 1      (45,37) 1      (48,44) 1      (17,50) 1  
(35,2) 1      (49,19) 1      (30,32) 1      (49,37) 1      (50,44) 1      (21,50) 1  
(36,2) 1      (51,20) 1      (31,32) 1      (52,37) 1      (6,45) 1      (44,50) 1  
(29,3) 1      (54,20) 1      (33,32) 1      (56,37) 1      (18,45) 1      (48,50) 1  
(30,4) 1      (60,20) 1      (34,32) 1      (62,37) 1      (32,45) 1      (51,50) 1  
(31,4) 1      (50,21) 1      (37,32) 1      (10,38) 1      (37,45) 1      (52,50) 1  
(32,4) 1      (51,21) 1      (41,32) 1      (11,38) 1      (41,45) 1      (20,51) 1  
(33,4) 1      (52,21) 1      (45,32) 1      (36,38) 1      (46,45) 1      (21,51) 1  
(34,4) 1      (53,22) 1      (49,32) 1      (37,38) 1      (49,45) 1      (50,51) 1  
(31,5) 1      (55,22) 1      (52,32) 1      (39,38) 1      (52,45) 1      (52,51) 1  
(32,6) 1      (59,22) 1      (56,32) 1      (10,39) 1      (56,45) 1      (54,51) 1  
(37,6) 1      (61,22) 1      (62,32) 1      (12,39) 1      (58,45) 1      (60,51) 1  
(41,6) 1      (53,23) 1      (4,33) 1      (36,39) 1      (62,45) 1      (6,52) 1  
(45,6) 1      (54,23) 1      (7,33) 1      (37,39) 1      (15,46) 1      (21,52) 1  
(49,6) 1      (55,24) 1      (30,33) 1      (38,39) 1      (18,46) 1      (32,52) 1  
(52,6) 1      (56,24) 1      (31,33) 1      (40,39) 1      (42,46) 1      (37,52) 1  
(56,6) 1      (57,24) 1      (32,33) 1      (12,40) 1      (43,46) 1      (41,52) 1  
(62,6) 1      (57,25) 1      (34,33) 1      (13,40) 1      (45,46) 1      (45,52) 1  
(33,7) 1      (58,26) 1      (4,34) 1      (39,40) 1      (47,46) 1      (49,52) 1  
(34,8) 1      (59,27) 1      (8,34) 1      (6,41) 1      (58,46) 1      (50,52) 1  
(35,9) 1      (60,27) 1      (30,34) 1      (14,41) 1      (15,47) 1      (51,52) 1  
(36,10) 1      (61,28) 1      (31,34) 1      (32,41) 1      (19,47) 1      (56,52) 1  
(37,10) 1      (62,28) 1      (32,34) 1      (37,41) 1      (42,47) 1      (62,52) 1  
(38,10) 1      (2,29) 1      (33,34) 1      (42,41) 1      (43,47) 1      (22,53) 1  
(39,10) 1      (3,29) 1      (2,35) 1      (45,41) 1      (46,47) 1      (23,53) 1  
(38,11) 1      (30,29) 1      (9,35) 1      (49,41) 1      (48,47) 1      (54,53) 1  
(39,12) 1      (35,29) 1      (29,35) 1      (52,41) 1      (49,47) 1      (55,53) 1  
(40,12) 1      (36,29) 1      (30,35) 1      (56,41) 1      (17,48) 1      (59,53) 1  
(40,13) 1      (2,30) 1      (36,35) 1      (62,41) 1      (19,48) 1      (61,53) 1  
(41,14) 1      (4,30) 1      (2,36) 1      (14,42) 1      (44,48) 1      (20,54) 1  
(42,14) 1      (29,30) 1      (10,36) 1      (15,42) 1      (47,48) 1      (23,54) 1  
(42,15) 1      (31,30) 1      (29,36) 1      (41,42) 1      (49,48) 1      (51,54) 1  
(43,15) 1      (32,30) 1      (30,36) 1      (43,42) 1      (50,48) 1      (53,54) 1  
(46,15) 1      (33,30) 1      (35,36) 1      (46,42) 1      (6,49) 1      (60,54) 1  
(47,15) 1      (34,30) 1      (37,36) 1      (47,42) 1      (19,49) 1      (22,55) 1  
(43,16) 1      (35,30) 1      (38,36) 1      (15,43) 1      (32,49) 1      (24,55) 1  
(44,16) 1      (36,30) 1      (39,36) 1      (16,43) 1      (37,49) 1      (53,55) 1  
(44,17) 1      (4,31) 1      (6,37) 1      (42,43) 1      (41,49) 1      (56,55) 1  
(48,17) 1      (5,31) 1      (10,37) 1      (44,43) 1      (45,49) 1      (57,55) 1  
(50,17) 1      (30,31) 1      (32,37) 1      (46,43) 1      (47,49) 1      (59,55) 1  
(45,18) 1      (32,31) 1      (36,37) 1      (47,43) 1      (48,49) 1      (61,55) 1  
(46,18) 1      (33,31) 1      (38,37) 1      (16,44) 1      (52,49) 1      (6,56) 1  
(58,18) 1      (34,31) 1      (39,37) 1      (17,44) 1      (56,49) 1      (24,56) 1
```

(32,56) 1	(62,56) 1	(46,58) 1	(27,60) 1	(59,61) 1	(49,62) 1
(37,56) 1	(24,57) 1	(22,59) 1	(51,60) 1	(62,61) 1	(52,62) 1
(41,56) 1	(25,57) 1	(27,59) 1	(54,60) 1	(6,62) 1	(56,62) 1
(45,56) 1	(55,57) 1	(53,59) 1	(59,60) 1	(28,62) 1	(61,62) 1
(49,56) 1	(56,57) 1	(55,59) 1	(22,61) 1	(32,62) 1	
(52,56) 1	(18,58) 1	(60,59) 1	(28,61) 1	(37,62) 1	
(55,56) 1	(26,58) 1	(61,59) 1	(53,61) 1	(41,62) 1	
(57,56) 1	(45,58) 1	(20,60) 1	(55,61) 1	(45,62) 1	

2. RECONSTRUCTION OF **NAMES.mat** , size 62x12

```
NAMES=char('!','sow','old','pig','three','first','second','third','poor',
'sent','away','seek','fortune','meet','man','bundle','straw','say',
'give','house','build','wolf','come','ask','enter','no','blow','eat',
'sow old','sow pig','pig three','first pig','second pig','third pig','sow poor',
'sow sent','sent first','sent away','sent seek','seek fortune','first meet','meet
man','man bundle','bundle straw','first say','say man','man give','give straw',
'give first','straw build','build house','first build','wolf come','come house',
'wolf ask','ask first','ask enter','say no','wolf blow','blow house','wolf eat',
'eat first');
```

Part 2: SCALE

1. SCALE

```
% PROGRAM SCALE, script sc; RELATED: link;
```

```
% 1. INITIALIZE
```

```
S=input('1 to start, 2 to continue ');
nn=input('enter number of cycles (nn): ');
disp(' ATTENTION: Interspaced components should be entered')
disp(' between apostrophes as single string of characters');
disp(' ');
disp(' press any key to continue'); pause;
```

```
% 2. START
```

```
for n=1:nn, if (S==1)&(n==1), sw=1; W=zeros(sw, 16,8);
    NAMES=['!']; WW=zeros(sw,sw);NM=[ ];
    end
```

% 3. INPUT

```
I=input('enter components ');
```

% 4. CONVERT INPUT into PIN;

```
PIN=zeros(8,8); word=[];jj=1; I=[I, '']; ab=abs(I); lab=length(ab);  
for i=1:lab, if (ab(i)~=32), word=[word,ab(i)]; else  
lw=length(word); PIN(jj,1:lw)=word; word=[]; jj=jj+1; end, end
```

% 5. CHECKING novelty of PIN against WORLD

```
old=0; wiold=0; comp=[]; % PIN presumed new
```

```
for wi=1:sw %wi: length of World
```

```
% 5.1 CUTTING FLAT SLICE OF WORLD W(wi,,:)
```

```
fW=zeros(16,8); fW(:,:)=W(wi,,:);
```

% 5.2 COMPARE flat W and PIN as ordered sets %

```
eqid=isequal(fW(9:16,:),PIN); if eqid==1, old=1; %PIN is OLD  
disp('This is old '),disp(NAMES((wi,:)), % wiold=wi; list=find(WW(:,wiold));  
leli=length(list); SN=[];for k=1:leli, SN=strvcat(SN,NAMES((leli,:)));  
end  
end
```

% 5.3 COMPARE flat W and PIN as sets

```
set=0;  
if isempty(setxor(fW((1:8),:),PIN,'rows')),if old==0,  
if eqid~=1,  
note = ['Looks like W' ,int2str(wi),': ', NAMES(wi,:)];  
disp(note); set=1; end, end,old=1; end  
%%  
%%  
%%
```

```
if set==1,  
nnew=input(' Is it new? 1/0 ');
```

```
if nnew==1,old=0; end, end
```

% 5.4. CHECK for SET MEMBERS to create SPECTRUM;

```
eqcomp = intersect(PIN,fW ,'rows'); h=size(eqcomp,1) ; test=zeros(1,8);
```

```
for hh=1:h,
```

```
if ((~isempty(eqcomp))&(~isequal(eqcomp(hh,:),test))),
```

```
comp=[comp,wi];end, end
```

```
end %for wi=1:sw
```

```
spectrum=unique(comp);les=length(spectrum);
```

```
%disp('spectrum: '); %for r=1:les, disp(NAMES(spectrum(r,:))); end
```

```
%if old==1,SNU=unique(SN,'rows');disp('OLD spectrum '),disp(SNU),end,
```

% 6. IF NEW, EXPAND THE WORLD:

% 6.1. NAME THE NEW ENTRY

```
if old==0, NAME=input('This is new. Name it ');
```

```
NAMES=strvcat(NAMES, NAME);
```

% 6.2. CREATE NewPIN containing NAME

```

NAME=[NAME, ' ']; NewPIN=zeros(8,8);word=[];jj=1; ab=abs(NAME);
lab=length(ab); for i=1:lab, if (ab(i))~=32, word=[word,ab(i)];else
    lw=length(word); NewPIN(jj,1:lw)=word; word=[];jj=jj+1; end,
end

```

% 6.3.ADD PIN and NewPIN to WORLD

```

sw=sw+1; WW(sw,:)=0; WW(:,sw)=0;% PLACE IN THE WORLD
W(sw,9:16,:)=NewPIN(:,:); W(sw,1:8,:)=PIN(:,:);
% All 16 lines in WORLD are filled up

```

% 6.4. RECORD NEW LINKS

```

for ll=1:les, if ~isempty(spectrum(ll)),
    WW(sw,spectrum(ll))=1; WW(spectrum(ll),sw)=1;end, end,
end

```

```

% if old==0,

```

```

%To read an entry from W(x,y,:)

```

```

%D=nonzeros(W(x,y,:)); D=D'; setstr(D);

```

```

end % for n=1:nn

```

```

NM=NAMES; world=WW;

```

```

disp('If you want to check WORLD for a name, type: link ')

```

```

disp('To see NAMES, type: NAMES or NM ');

```

```

disp('To display the 3D world, type plotw ');

```

2. LINK

% CHECKING THE WORLD FOR NAME; SCRIPT: link

```

I=input('NAME to check ');

```

```

CI=cellstr(I);

```

```

SC=nonzeros(strmatch(CI,NAMES));%find NAME's numbers

```

```

LINKS=find(WW(:,SC)); links=LINKS';

```

```

LL= [ ]; %list of links

```

```

NAMELIST=[];

```

```

for i=1:length(SC)

```

```

    LL= [LL,(find(WW(:,SC(i))))]';

```

```

end

```

```

lel=length(LL);

```

```

for r=1:lel, NAM=(NAMES(LL(r,:)); NAMELIST=strvcat(NAMELIST,NAM);end,

```

```

NAMELIST=unique(NAMELIST,'rows'); disp(NAMELIST)

```

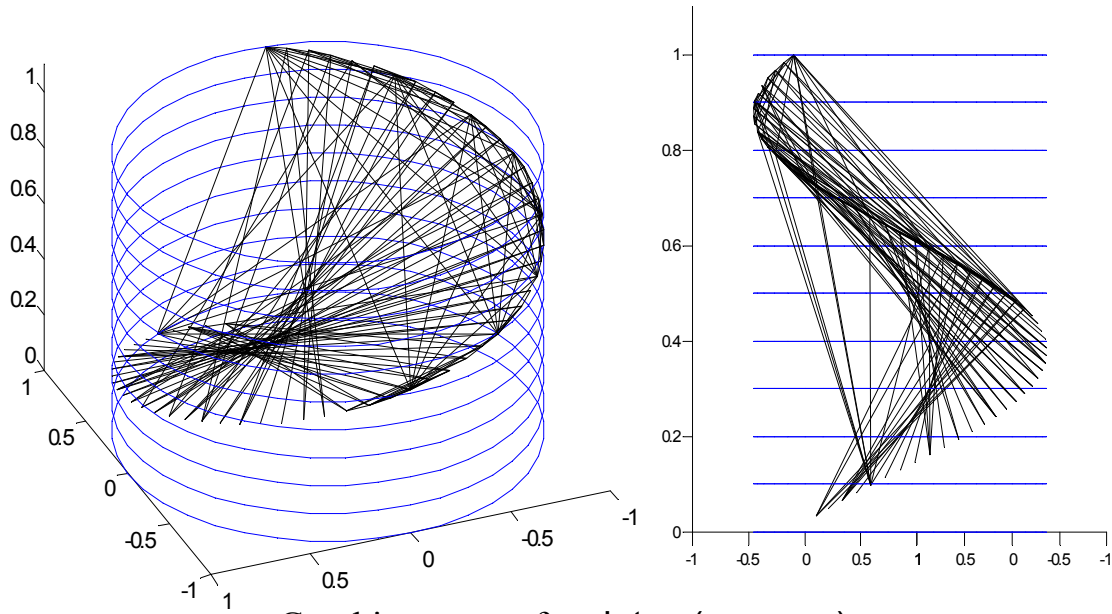
3. PLOTW

```

%script: plotW, plot 3D WORLD
%tic
axis([-1.2, 1.2, -1.2, 1.2, 0, 1.2]); stack, om=length(WW);
PX=zeros(om,1); PY=zeros(om,1); PZ=[1:om]; PZ=PZ/om;
angles=(2*pi/om).*[0:om];
for j=1:om, PX(j)=cos(angles(j)); PY(j)=sin(angles(j)); end

cor=zeros(om,4); cor(:, 1) = [1:om]'; cor(:, 2)=PX; cor(:, 3)=PY; cor(:, 4)=PZ;
for j = 1: om,
    fw= find(WW(j,:)); lfw=length(fw);
    for k=1: lfw
        kk=fw(k); if kk <=j ,
            X= [cor(j, 2) cor(kk,2)]; Y= [cor(j, 3) cor(kk,3)]; Z= [cor(j, 4) cor(kk,4)];
            pause(0.1); plot3 (X,Y,Z,'Color','black','LineWidth', 1 );
        end,
    end, pause(0.1);
end
end
%toc

```



Graphic output of plotw (animated)

Part 3: EVE

1. INIT

```
%open workspace
om=length(WW); P (1 : om, 3 ) = 1/om ;
angles=(2*pi/om).*[0:om-1];
% AR: arity
%AR=0;for i = 1:om , line = find ( WW ( i , : ) ); sl=length (line); AR(i)= sl; end
%for k=1:om, NUMBERS(k,1)= k , end;
```

2. P3D3

%PIG, script p3d3.m,

```
axis ([-1.1, 1.1, -1.1, 1.1, 0, 1.1]); hold on
%DISPLAY NUMBERS open p3d62.fig
```

```
LC=[1 1 0];
```

```
%t = input ( 'Enter pause between the cycles t ' ) % t=0.1 sec
tra=[]; % tra: trajectory
om=length(WW); AA=zeros(n,1); % AA contains subsequent actons
ain=a; %ain: storage of the initial acton;
PZ=[1:om]; PZ=PZ/om;
```

%%%%ARITY CALCULATION

```
%%%%%%for y=1:62, lfw=length(find(WW(y,:))); P(y,4)=lfw; end
```

%DISPLAY PARAMETERS IN FIGURE

```
%PAR12=['G=',num2str(G)];text (-1.2,-1.2, 1, PAR12); PAR12=['H=',num2str(H)];text
(-1.2,-1.2, 1.2,PAR12); PAR12=['C=',num2str(C)];text ( -1.2,-1.2, 1,PAR12);
PAR12=['F=',num2str(F)];text ( -1.2,-1.2, 0.9,PAR12); PAR12=['a=',num2str(aa)];text
(-1.2,-1.2, 0.8,PAR12); PAR12=['acton =',NAMES(aa,:)];text ( -1.2,-1.2, 0.7,PAR12);
%PAR12=['n=',num2str(n)]; text ( -1.2,-1.2 ,0.6,PAR12);
```

%%%%%%PURGE INITIAL PROBABILITIES P.

```
% P (1 : om, ) = 1/om ; % Equal probabilities.
% or : use script "init"
```

```

%%%% PREVIOUS ACTON DATA
P(1,3)=0; % empty word '!' cannot be selected
P(ain,3)=0; pra=ain; %pra : previous acton

% DISPLAY ain as green circle
text (0.9*(cos(angles(ain))), 0.9*(sin(angles(ain))), 0, 'o','color','green','FontSize',16 );

for jj=1:n % MAIN LOOP BEGINS

% START ACTON SELECTION
select3

% END ACTON SELECTION

% ACTON ASTERISK DISPLAY; Randomization of the position of ASTERISK.

%RANDOMIZATION OF SIGN
u=rand/30; uu=rand; if uu<=0.5, uuu=1; else uuu=-1; end, u=u*uuu;
v=rand/30; vv=rand; if vv<=0.5, vvv=1; else vvv=-1; end, v=v*vvv;

%text (( cos ( angles (a))+u ) , ( sin (angles(a))+v ) , '*' , 'color' , [1 0 0] , 'FontSize' , 18);
%pause(t),

text((cos(angles(a))+u),(sin(angles(a))+v),PZ(a), '*' , 'color' , 'red' , 'FontSize' , 18);
pause(0.1),
%text ( 0.8*( cos ( angles (a))+u ) , 0.8*(sin (angles(a))+v ) , '*' , 'color' , 'green' ,
'FontSize' , 18),
%!!!!!!!!!!!!!!!!!!!!!!!!

LLX= cos(angles(a))+u;
LLY= sin(angles(a))+v;
LLZ=PZ(a); LLC=[LLX LLY LLZ] ;
axis ([-1.2, 1.1, -1.1, 1.1, 0, 1.1]);
X1=LC(1); X2=LLC(1); Y1=LC(2); Y2=LLC(2); Z1=LC(3); Z2=LLC(3);
X=[X1 X2]; Y=[Y1 Y2]; Z=[Z1 Z2]; hold on,
plot3(X,Y,Z,'Color','red','LineWidth', 1 );
LC=LLC ;

%END ASTERISK DISPLAY

% END OF PROTO FOLLOWS

AA ( jj , 1) = a;
end

tra = AA';

```

```
PP=zeros(om,2); PP(:,1) = P( 1 : om , 1); PP ( : , 2 ) = P ( 1 : om , 3);
```

```
%disp ('type PP for probability distribution ');  
%disp ( 'type tra for trajectory' );  
%disp(' '), disp (['total content ', num2str(ld)] );  
%disp (['condensate size ', num2str(ld)] );  
%disp (['JINI = ', num2str(JINI)]);
```

```
stack
```

3. SELECT3

```
%ACTON SELECTION
```

```
%START PROBABILITY DISTRIBUTION
```

```
pra=ain;
```

```
P(pra,3) = H*C ; % influence of former acton on neighbors; for calculating influence  
only
```

```
for k=1:om,  
    INF=0; line = find ( WW ( k , : ) ); sl=length (line); ar= sqrt(sl);  
    for j=1:sl, neighbor=line(j); INF=INF+ G*P(neighbor, 3)/ar; end,
```

```
%INFLUENCE
```

```
P(k,3)=P(k,3)- F * P(k,3 ) +INF;
```

```
end
```

```
P(a,3)= 0; P(1,3)=0; %P(a,3) reverted to 0
```

```
%END PROBABILITY DISTRIBUTION
```

```
% START ACTON SELECTION
```

```
    %P(a,3)=0; %cannot be selected again
```

```
    pra=a; % previous acton "a" remembered as "pra"
```

```
    P(1,3)=0; SUM=0; ss=0; S=P(1:om , 3); SUM=sum(S);
```

```
    P(1:om , 3)=P(1:om,3)/SUM;
```

```
    %probabilities are normalized
```

```
    r=rand; for i=1:om, ss=ss+P(i,3); if ss>r, a=i; break, end
```

```
    end %NEW ACTON "a" HAS BEEN SELECTED
```

```
    P(pra,3)=C; %new probability for former acton; now it can take part in next selection
```

```
    %P(1,3)=0; % '1' and acton cannot be selected
```

```
% END ACTON SELECTION
```

4. NUMS

%%NUMERICAL ANALYSIS AND JINI

%nums calculates occurrence of a cell in the trajectory and Jini coeff.

% N(:,1) lists cells, N(:,2) lists occurrence

%open NAMES.mat;

DDA=[]; srt=sort(AA);

ls=length(srt); lss=ls-1; U=unique(AA); lu=length(U); N=zeros(lu,2);

N(:,1)=U; j=1; VV=0;

for k=1:lss, %over srt, except the last

if srt(k)==srt(k+1), VV=VV+1; else N(j,2)=VV+1; j=j+1; VV=0; end

end,

N(lu, 2)= VV+1;

ns=N(:,2)'; ns=sort(ns); lns=length(ns);

lms=length(ns); JIN=[1:lns]; JIN(1)=ns(1);

for k=2:lns, JIN(k)=JIN(k-1)+ ns(k); end,

SJ=sum(JIN); SE= (JIN(lns)*lu)/2; JINI=SJ/SE;

D=sortrows(N,2); NMS=NAMES(D(:,1),:); ld=length(D);

%D, lu, 'ns', ns', 'JIN', JIN', JINI

%N

%JIN

% ddn OCCURENCE DISTRIBUTION

disp(' '); disp ('content:'); disp(' ');

DDA= D(:,1);

%base=D(:,1)'; NAMES(base,:) %display content names only

ldda=length (DDA);

%DISPLAY WORD No, OCCURRENCE, WORD

for i = 1:ldda , disp ([num2str(i), ' ', num2str(D(i,1)),' ', num2str(D(i,2)),' ' ,

NAMES(D(i,1), :)]), end

%for i = 1:ldda , disp ([num2str(D(i,2)),' ' , NAMES(D(i,1), :)]), end

cont=D(:,1); content=sort(cont);

%for f=1:ldda, disp ([num2str(TT(f)),' ', num2str(TT(f,2)),' ' , NAMES(AA(f,1), :)]),

end

%disp(' '), disp (['total content ' , num2str(ld)]); disp (['JINI = ' , num2str(JINI)]);

5. SAYCON2

```
% Program saycon2 as script, can be used as function(content,connector)
% saycon = "say_con(content,connector)"
```

%Part 1: SPLIT: splcomplex cells

```
    %lists content , builds structure SPLIT: name, nleft, nright
    %splits composite name into component NAMES

wl=[];wm=[];wr=[];d=1;
lcnt=length(content); SPLIT(1).name=[]; SPLIT(1).nleft=[]; SPLIT(1).nright=[];

    for k=1:lcnt, SPLIT(k).name= cellstr(NAMES(content(k,:)));
    end;
% SPLIT( : , : ).name generates list of names;
fst=[]; %for find str
    %sspl=size(SPLIT); ssp=sspl(2);
csp=0;

for k=1:lcnt, fst= findstr (' ', char( SPLIT(k).name) ); %find space between components
    %if no space:
    if isempty(fst)==1, SPLIT(k).nleft=char( SPLIT(k).name); SPLIT(k).nright=[]; %only
left component
    %if space
    else csp=char( SPLIT(k).name); lcssp=length(csp); %total NAME
        SPLIT(k).nleft=csp(1:(fst-1)); SPLIT(k).nright =csp((fst+1):lcssp);fst=[];
    end,
end
```

%Part 2: CONDENSATE

```
disp (' '); disp('condensate:'); disp (' ');

    TRIPLET(1).L=[]; TRIPLET(1).R = []; TRIPLET(1).M =[];
    TRIPLET(1).TRI=[];
    TRI1 = ['_','_','_','_','_','_','_'];
    nt=0; nnt=0;
for c=1:lcnt,

    for m=(1:lcnt),
```

```

dd= isequal( SPLIT(c).nleft, SPLIT(m).nright);
    if dd==1,      d=d+1;
    wl=SPLIT(m).nleft; wr=(SPLIT(c).nright); wm =(SPLIT(c).nleft); else

    TRI2= [wl,'_',wm,'_', wr, ' ; '];
    end,

tt=isequal (TRI1,TRI2); % if not and if wr not empty :
    if tt==0,  ttt=isempty(wr) ;
        if ttt==0,
            NC=[wl,' ', wm,]; NM=[wm,' ', wr,];
            %NC=char( SPLIT(c).name);
            [tfc, locc] =ismember ( NC, NAMES, 'rows');
            %NM=char( SPLIT(m).name);
            [tfm, locm] =ismember ( NM, NAMES, 'rows');

            if abs(locc-locm)<=5,

% 5 is the range of narrow condensate

% NARROW CONDENSATE IS MARKED WITH ASTERISK
            nnt=nnt+1; disp(['TRI2, ' * ']), else
            disp (TRI2),
            %disp([ wl,'_',wm,'_', wr, ' ; ']) ,
            end
            nt=nt+1;
            end,

            TRI1=TRI2;
            end, %if tt==0,

    TRIPLET(d).L =wl; TRIPLET(d).R = wr; TRIPLET(d).M = wm; %for a record
    TRIPLET(d).TRI = [TRIPLET(d).L,'_',TRIPLET(d).M,'_', TRIPLET(d).R, ' ; '];

    end, %for m=(1:lcnt)

end %for c=1:lcnt,

disp(' '), disp(['total content ', num2str(ld)] );
disp(['condensate ', num2str(nt)];disp(['narr. cond.', num2str(nnt)]);
disp(['JINI = ', num2str(JINI)]; disp(['cond/cont = ', num2str(nt/ld)]);
disp(['nr.cnd/cnt = ', num2str(nnt/ld)]);

```



```

for m=((mm+1):lcnt),

dd= isequal( SPLIT(k).nleft, SPLIT(m).nright);
if dd==1,    d=d+1;
wl=SPLIT(m).nleft; wr=(SPLIT(k).nright); wm =(SPLIT(k).nleft); else

TRI2= [wl,'_',wm,'_', wr, ' '];
end,

tt=isequal (TRI1,TRI2); if tt==0, ttt=isempty(wr) ; if ttt==0,    disp(TRI2);nt=nt+1;
    %%%%%%%%%%%
TRI1=TRI2; end, end
TRIplet(d).L =wl; TRIplet(d).R = wr; TRIplet(d).M = wm; %for a record
TRIplet(d).TRI = [TRIplet(d).L,'_',TRIplet(d).M,'_', TRIplet(d).R, ' '];

end, end
disp(' '), disp(['total content ', num2str(ld)] );
disp(['condensate ', num2str(nt)])
disp(['JINI = ', num2str(JINI)]); disp(['cond/cont = ', num2str(nt/ld)]);

```

6. STACK

```

%script: stack
ZZ=zeros(1,41); tt=0;
for zz=0:0.1:1
    t = 0:pi/20:(2*pi); ZZ(1,:)=zz; tt=ZZ;
    plot3(sin(t),cos(t),tt), hold on, tt=tt+1;
end, axis ([-1, 1, -1, 1, 0, 1.1]);

```

7. TR

```

%tr  replays TRAJECTORY as NAMES
% compare with tr3d, which replays 3D TRAJECTORY IN TIME
%compare with repro: 3D TRAJECTORY IN SPACE ; replays AA trajectory in
WORLD

%NUMBERS AND WORDS
ltra=length (AA); for i = 1:ltra  ntr = AA(i); disp ([num2str(i),' ', num2str(ntr),' ',
NAMES(ntr,:)]), end

%NUMBERS
% ltra=length (tra); for i = 1:ltra  ntr = tra(i); disp ([num2str(i),' ', num2str(ntr)]), end

```

```

% TRAJECTORY TO SAVE AS TT
T=zeros(2,n); ltra=length (AA); ntr=0; for i = 1:ltra  ntr = AA(i); T(1,i)= i; T(2,i) = ntr;
end, TT= T';

```

8. TR3D

```

% tr3d : replays AA as 3D TRAJECTORY IN TIME
% compare with tr : replays TRAJECTORY as NAMES
%compare with repro:
%3D TRAJECTORY IN SPACE ; replays AA trajectory in WORLD

```

```

n=length(AA); om=length(WW);
h=2; %height
figure, axis ([-1.4, 1.4, -1.4, 1.4, 0, 2.1]); view(-160, 55),hold on,
%DISPLAY NUMBERS
for zz=0:0.2:h, t = 0:pi/20:(2*pi); ZZ(1,:)=zz; tt=ZZ;
    plot3(sin(t),cos(t),tt), hold on, tt=tt+1; end

```

```

zz=2/n;
X0= [cos(angles(ain)) cos(angles(AA(1)))];
Y0=[sin(angles(ain)) sin(angles(AA(1)))];
Z0=[0 zz];

```

```

plot3(X0,Y0,Z0,'Color','black','LineWidth', 1 );

```

```

pause(0.1);
nn=(n-1);
for k=(1:nn),

```

```

X= [cos(angles(AA(k))) cos(angles(AA(k+1)))];
Y= [sin(angles(AA(k))) sin(angles(AA(k+1)))];
Z= [zz*(k) zz*(k+1) ];

```

```

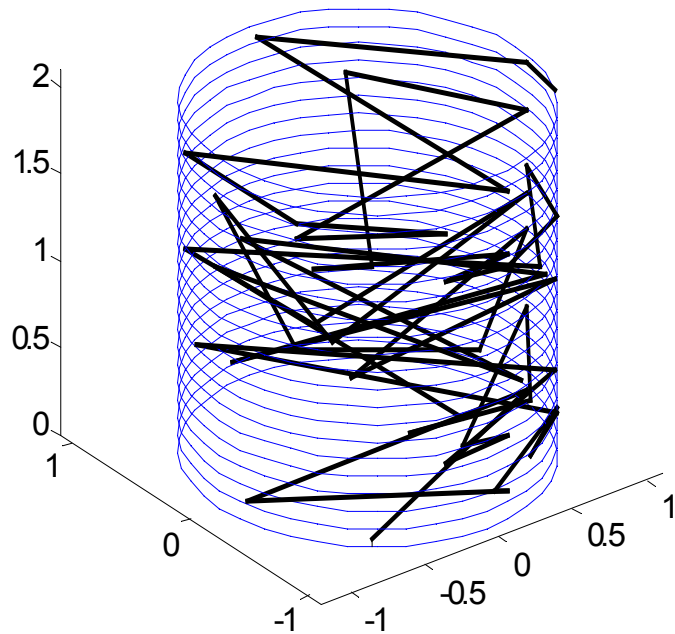
plot3 (X,Y,Z,'Color','black','LineWidth', 2 ); hold on

```

```

pause(0.1);
end
for j=1:om, text ( 1.15*cos ( angles (j) ) , 1.15*sin (angles(j)) , num2str(j),'FontSize', 8),
end

```



Graphic output of tr3d

9. REPRO replays AA (trajectory)

EXAMPLE OF INPUT:

ENTER:

```
AA=[ 41 62 34 49 48 57 51 55 21 58 45 49 46 62 51 43 46 19 50 17 60 33
58 40 62 47 37 19 39 62 38 28 52 62 56 49 45 49 38 41 11 62 37 45 37 28
49 16 62 58]; repro;
[ENTER]
```

REPRO.m

```
%repro: 3D TRAJECTORY IN SPACE ; replays AA trajectory in WORLD
% compare with tr3d, which replays 3D TRAJECTORY IN TIME
%compare with tr, which replays TRAJECTORY as NAMES

cd C:\MATLAB6p5\work\PIG; load NAMES.mat, load WW.mat,
n=length(AA);ain=AA(1);nn=n-1; om=length(WW); zz=1/om;

angles=(2*pi/om).*[0:om-1]; ZV=[0 : zz : 1];%COORDINATES
```

```
omm=om-1; figure, stack, axis ([-1.2, 1.2, -1.2, 1.2 , 0, 1.1]); view(-160, 55),
```

```
%DISPLAY NUMBERS
```

```
%for j=1:om, text ( 1.1*cos ( angles ( j ) ) , 1.1*sin ( angles(j)) , num2str(j),'FontSize', 8),  
end
```

```
%"ain" display
```

```
text (0.9*(cos(angles(ain))), 0.9*(sin(angles(ain))), 'o','color', 'green','FontSize',20 );
```

```
%ASTERISK AND LINE DISPLAY;
```

```
for jj=1:nn ; a1=AA(jj); a2=AA(jj+1);
```

```
    % Randomization of the position of ASTERISK.
```

```
        u=rand/25; uu=rand; if uu<=0.5, uuu=1; else uuu=-1; end, u=
```

```
        v=rand/25; vv=rand; if vv<=0.5, vvv=1; else vvv=-1; end, v=v*vvv;
```

```
        w=rand/25; ww=rand; if ww<=0.5, www=1; else www=-1; end, w=w*www;
```

```
    %ASTERISK
```

```
        text (0.9*(cos(angles(a2))+u), 0.9*(sin(angles(a2))+v), (ZV(a2)+w),'*','color',  
'red','FontSize',14 );
```

```
        pause(0.1), hold on %
```

```
%LINE
```

```
        X1=0.9*cos(angles(a1))+u; Y1=0.9*sin(angles(a1))+v; Z1=ZV(a1)+w;
```

```
        X2=0.9*cos(angles(a2))+u; Y2=0.9*sin(angles(a2))+v; Z2=ZV(a2)+w;
```

```
        X=[X1 X2]; Y=[Y1 Y2]; Z=[Z1 Z2];
```

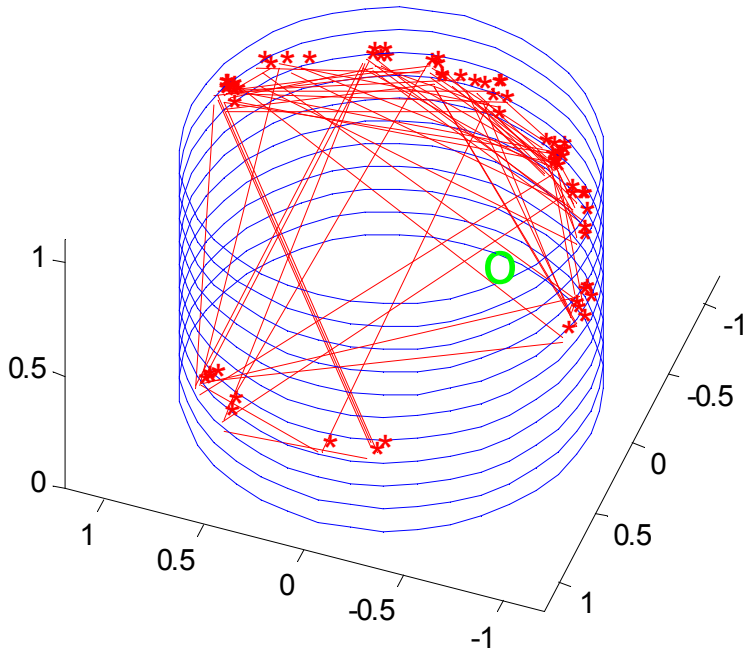
```
        plot3(X,Y,Z, 'Color','red','LineWidth',1,'LineStyle','-'); hold on,
```

```
% END ASTERISK AND LINE DISPLAY
```

```
end %jj;
```

```
disp([' n: ', num2str(n), ' initial acton: ', num2str(ain)] )
```

```
disp('run tr for TRAJECTORY with names, type tr3d, for 3D TRAJECTORY')
```



Graphic output of repro

10. REPRO2D , requires WW.mat and NAMES.mat

```
%repro2d: 2D distribution of actons over NAMES
% compare with tr3d, which replays 3D TRAJECTORY IN TIME
%compare with tr, which replays TRAJECTORY as NAMES
```

```
n=length(AA);ain=AA(1);nn=n-1; om=length(WW);
angles=(2*pi/om).*[0:om-1]; %COORDINATES
figure, axis([-1.2, 1.2, -1.2, 1.2 ]); hold on
t = 0:pi/20:(2*pi); plot(sin(t),cos(t),
```

```
%DISPLAY NUMBERS
```

```
for j=1:om, text ( 1.1*cos ( angles ( j ) ) , 1.1*sin ( angles(j)) , num2str(j),'FontSize', 8), end
```

```
%"ain" display
```

```
text ((cos(angles(ain))), (sin(angles(ain))), 'o', 'color', 'green', 'FontSize', 20 );
```

```
%ASTERISK AND LINE DISPLAY;
```

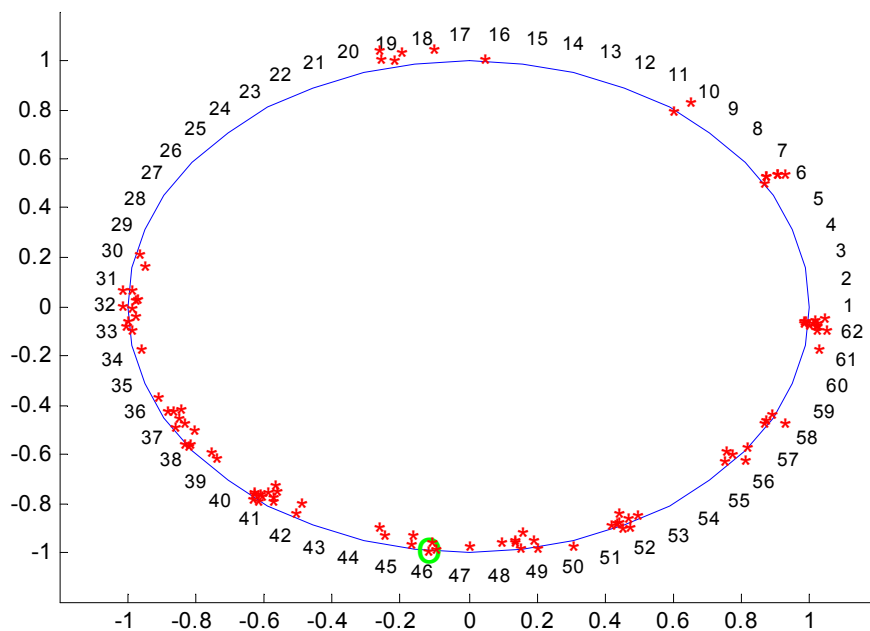
```
for jj=1:nn ; a1=AA(jj); a2=AA(jj+1);
```

```

% Randomization of the position of ASTERISK.
                                u=rand/25; uu=rand; if uu<=0.5, uuu=1; else uuu=-1; end, u=
v=rand/25; vv=rand; if vv<=0.5, vvv=1; else vvv=-1; end, v=v*vvv;
%ASTERISK
text ((cos(angles(a2))+u), (sin(angles(a2))+v), '*', 'color', 'red', 'FontSize', 14 );
pause(0.1), hold on %
%LINE , OPTIONAL
%X1=cos(angles(a1))+u; Y1=sin(angles(a1))+v; X2=cos(angles(a2))+u;
Y2=sin(angles(a2))+v;
%X=[X1 X2]; Y=[Y1 Y2]; plot(X,Y, 'Color','red','LineWidth',1,'LineStyle','-'); hold
on,

% END ASTERISK AND LINE DISPLAY
end %jj;
disp([' n: ', num2str(n), ' initial acton: ', num2str(ain)] )
disp('run tr for TRAJECTORY with names, type tr3d, for 3D TRAJECTORY')

```



Graphic output of repro2d

Part 4

Example of a batch command

```
>> C = 0.5; G = 1; H = 0.8; F = 0.8;
init, n=25
a=22; figure, stack, view(-160, 55),
p3d3, disp(' '); disp(['AA=[' , num2str(AA), ']; ']),
nums, say_con, disp(['ain ', num2str(ain), ' , a fin ' num2str(a)]),
```

```
n=25;
figure, stack, view(-160, 55),
p3d3, disp(' '); disp(['AA=[' , num2str(AA), ']; ']),
nums, content=D(:,1)'; say_con,
disp(['ain ', num2str(ain), ' , a fin ' num2str(a)])
```

Part 5 Structure of WORLD (62 x 62)

AR: Arity

No.	NAME	Neighbors (No.)	Neighbors (NAMES)	AR
1	!	0	!	0
2	sow	29, 30, 35, 36	sow old , sow pig , sow poor , sow sent	4
3	old	29	sow old	1
4	pig	30, 31, 32, 33, 34	sow pig , pig three , first pig , second pig , third pig	5
5	three	31	pig three	1
6	first	32, 37, 41, 45, 49, 52, 56, 62	first pig, sent first, first meet, first say , give first, first build, ask first, eat first	8
7	second	33	second pig	1
8	third	34	third pig	1
9	poor	35	sow poor	1
10	sent	36, 37, 38, 39	sow sent , sent first, sent away , sent seek ,	4
11	away	38	sent away	1
12	seek	39, 40	sent seek , seek fortune	2
13	fortune	40	seek fortune	1
14	meet	41, 42	first meet, meet man	2
15	man	42, 43, 46, 47	meet man , man bundle, say man , man give	4
16	bundle	43, 44	man bundle, bundle straw	2
17	straw	44, 48, 50	bundle straw, give straw, straw build ,	3
18	say	45, 46, 58	first say , say man , say no	3
19	give	47, 48, 49	man give , give straw , give first	3
20	house	51, 54, 60	build house , come house , blow house	3
21	build	50, 51, 52	straw build , build house , first build	3
22	wolf	53, 55, 59, 61	wolf come , wolf ask , wolf blow , wolf eat	4

23	come	53, 54	wolf come , come house	2
24	ask	55, 56, 57	wolf ask, ask first, ask enter	3
25	enter	57	ask enter	1
26	no	58	say no	1
27	blow	59, 60	wolf blow, blow house	2
28	eat	61, 62	wolf eat, eat first	2
29	sow old	2, 3, 30, 35, 36	sow , old , sow pig , sow poor , sow sent	5
30	sow pig	2, 4, 29, 31, 32 , 33, 34, 35, 36	sow , pig , sow old , pig three , first pig , second pig , third pig , sow poor , sow sent	9
31	pig three	4, 5, 30, 32, 33, 34	pig , three , sow pig , first pig , second pig , third pig	6
32	first pig	4, 6, 30, 31, 33, 34, 37, 41, 45, 49, 52, 56, 62	pig , first , sow pig , pig three , second pig , third pig , sent first , first meet , first say , give first , first build , ask first , eat first	13
33	second pig	4, 7, 30, 31, 32, 34	pig , second , sow pig , pig three , first pig , third pig	6
34	third pig	4, 8, 30, 31, 32, 33	pig , third , sow pig , pig three , first pig , second pig	6
35	sow poor	2, 9, 29, 30, 36	sow , poor , sow old , sow pig , sow sent	5
36	sow sent	2, 10, 29, 30, 35, 37, 38, 39	sow , sent , sow old , sow pig , sow poor , sent first , sent away , sent seek	8
37	sent first	6, 10, 32, 36, 38, 39, 41, 45, 49, 52, 56 12, 62	first , sent , first pig , sow sent , sent away , sent seek , first meet , first say , give first , first build , ask first , eat first	12
38	sent away	10, 11, 36, 37, 39	sent , away , sow sent , sent first , sent seek	5
39	sent seek	10, 12, 36, 37, 38, 40	sent , seek , sow sent , sent first , sent away , seek fortune	6
40	seek fortune	12, 13, 39	seek , fortune , sent seek	3
41	first meet	6, 14, 32, 37, 42, 45, 49, 52, 56, 62	first , meet , first pig , sent first , meet man , first say , give first , first build , ask first , eat first	10
42	meet man	14, 15, 41, 43, 46, 47	meet , man , first meet , man bundle , say man , man give	6
43	man bundle	15, 16, 42, 44, 46, 47	man , bundle , meet man , bundle straw, say man , man give	6
44	bundle straw	16, 17, 43, 48, 50	bundle , straw , man bundle , give straw , straw build	5
45	first say	6, 18, 32, 37, 41, 46, 49, 52, 56, 58, 62	first , say , first pig , sent first , first meet , say man , give first , first build , ask first , say no , eat first	11
46	say man	15, 18, 42, 43, 45, 47, 58	man , say , meet man , man bundle , first say , man give , say no	7
47	man give	15, 19, 42, 43, 46, 48, 49	man , give , meet man , man bundle , say man , give straw , give first	7
48	give straw	17, 19, 44, 47,, 49, 50	straw , give , bundle straw, man give , give first , straw build	6
49	give first	6, 19, 32, 37, 41, 45, 47, 48, 52, 56, 62	first , give , first pig , sent first , first meet , first say , man give , give straw , first build , ask first , eat first	11
50	straw build	17, 21, 44, 48, 51, 52	straw , build , bundle straw, give straw , build house , first build	6
51	build house	20, 21, 50, 52, 54, 60	house , build , straw build , first build , come house , blow house	6
52	first build	6, 21, 32, 37, 41, 45, 49, 50, 51, 56,	first , build , first pig , sent first , first meet , first say , give first , straw build , build	11

		62	house , ask first , eat first	
53	wolf come	22, 23, 54, 55, 59, 61	wolf , come , come house , wolf ask , wolf blow , wolf eat	6
54	come house	20, 23, 51, 53, 60	house , come , build house , wolf come , blow house	5
55	wolf ask	22, 24, 53, 56, 57, 59, 61	wolf , ask , wolf come , ask first , ask enter , wolf blow , wolf eat	7
56	ask first	6, 24, 32, 37, 41, 45, 49, 52, 55, 57, 62	first , ask , first pig , sent first , first meet , first say , give first , first build , wolf ask , ask enter , eat first	11
57	ask enter	24, 25, 55, 56	ask , enter , wolf ask , ask first	4
58	say no	18, 26, 45, 46	say , no , first say , say man	4
59	wolf blow	22, 27, 53, 55, 60, 61	wolf , blow , wolf come , wolf ask , blow house , wolf eat	6
60	blow house	20, 27, 51, 54, 59	house , blow , build house , come house , wolf blow	5
61	wolf eat	22, 28, 53, 55, 59, 62	wolf , eat , wolf come , wolf ask , wolf blow , eat first	6
62	eat first	6, 28, 32, 37, 41, 45, 49, 52, 56, 61	first , eat , first pig , sent first , first meet , first say , give first , first build , ask first , wolf eat	10

=====.

Last revised: November, 2010
[email](#)