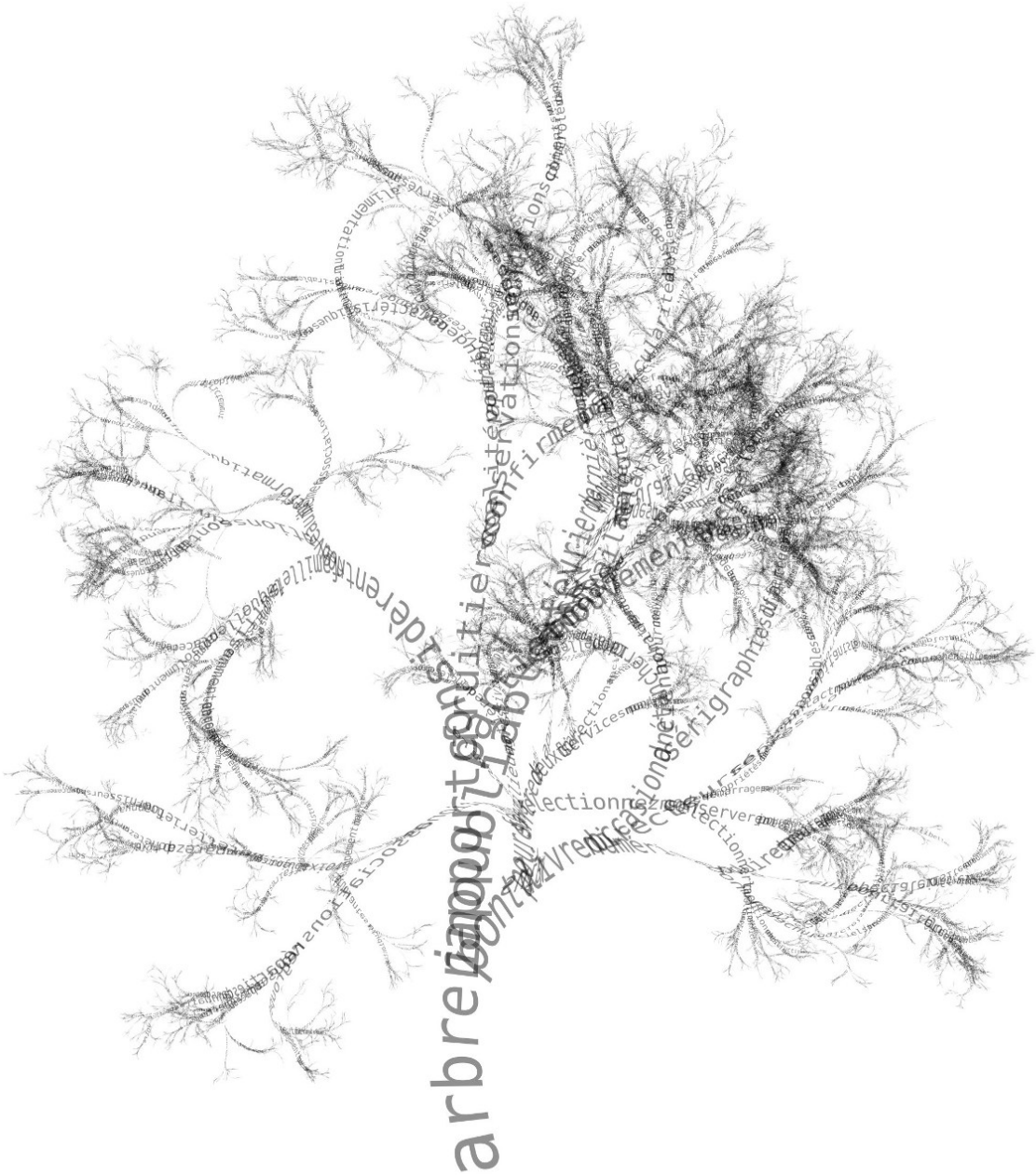


Ces arbres reposent sur une arborescence complexe composée des lettres de l’alphabet.
Cette arborescence, constituée de plusieurs dizaines de milliers de mots, s’alimente à partir d’un fichier texte. Chaque branche est générée à partir d’un mot choisi dans ce texte. Chaque occurrence de ce mot dans le texte est suivie par un autre mot, suite logique du texte, qui est utilisé pour former une branche suivante. L’arbre est donc généré selon un processus d’itération qui s’organise autour d’un contenu textuel. Chaque arbre dessine un chemin, un voyage possible à travers un texte. Les mots le constituant proviennent du champ lexical de l’auteur du texte.
Ces "Verbiages Végétaux" figurent ainsi une connectivité errante, une représentation fractale, une cartographie des possibilités infinies de navigation à travers la pensée d’un auteur, enchaînant un mot à un autre, une idée... Ils bousculent l’ordre chronologique, perturbent la syntaxe narrative. Ce sont des formes-forces décrivant un mouvement dans l’espace, déroulant une poétique du bruissement, des entrelacs et du hasard.
Ces œuvres interrogent la tension entre organisation et foisonnement, continuité et discontinuité, unité et multiplicité, simplicité et complexité...

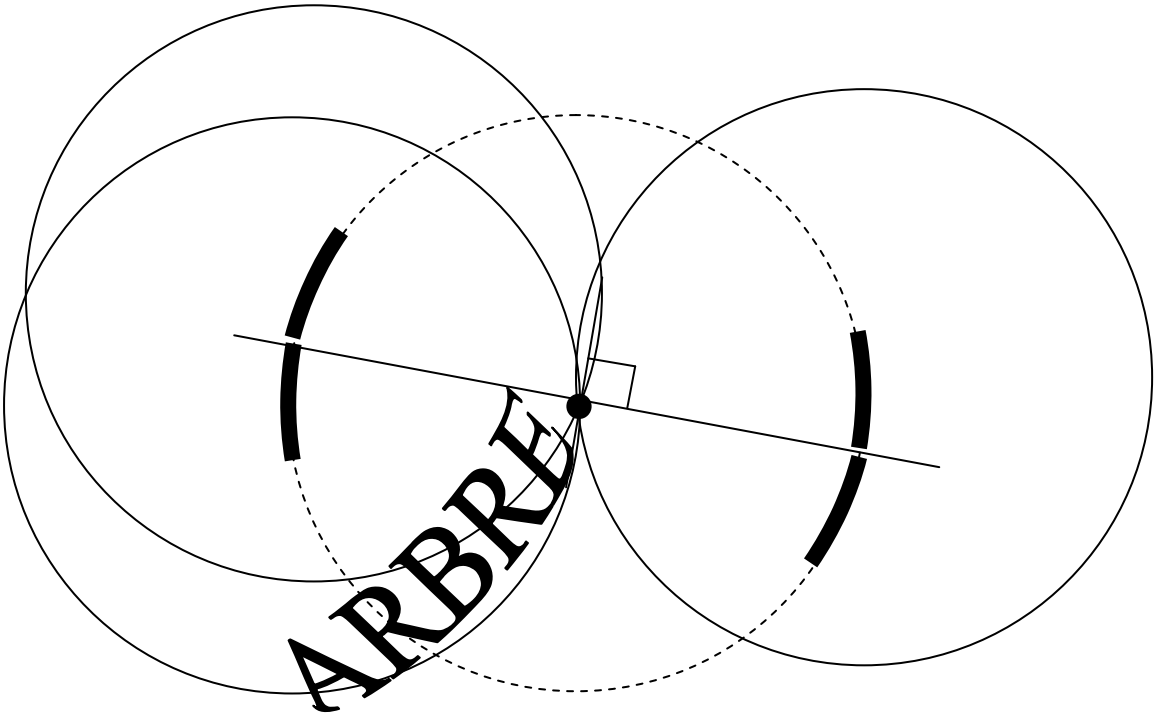
Verbiage Végétal, chnry, http://www.chnry.net/ch/

On ne conserve dans le texte que les mots ayant plus de trois occurrences.
A partir d'un premier mot, on regarde dans le texte, ces trois premières positions. On cherche ensuite trois autres mots qui suivent le premier mot et qui ont au moins trois occurrences. Ainsi de suite à partir de chaque nouveau mot.

Si le premier mot est d'une certaine taille, la taille de chaque autre mot choisi diminue d'une certaine proportion (10% ?).

A partir du pied du caractère qui devait suivre le premier mot, on fait partir les trois autres mots.

Quelle courbure prendre ? On pourrait considérer que les mots tracés se trouvent sur un cercle de rayon proportionnel à la taille de ce mot (rayon = entre 10 et 20 fois la taille de la police du mot ?)
Pour se faire, on trace, au pied du premier caractère qui devait suivre le mot, le cercle de centre ce point et de longueur la valeur attendue (proportionnelle à la taille de la police du mot …). On choisit les centres des trois cercles suivants proche de la perpendiculaire au mot précédent passant par le pied du caractère et tel que :

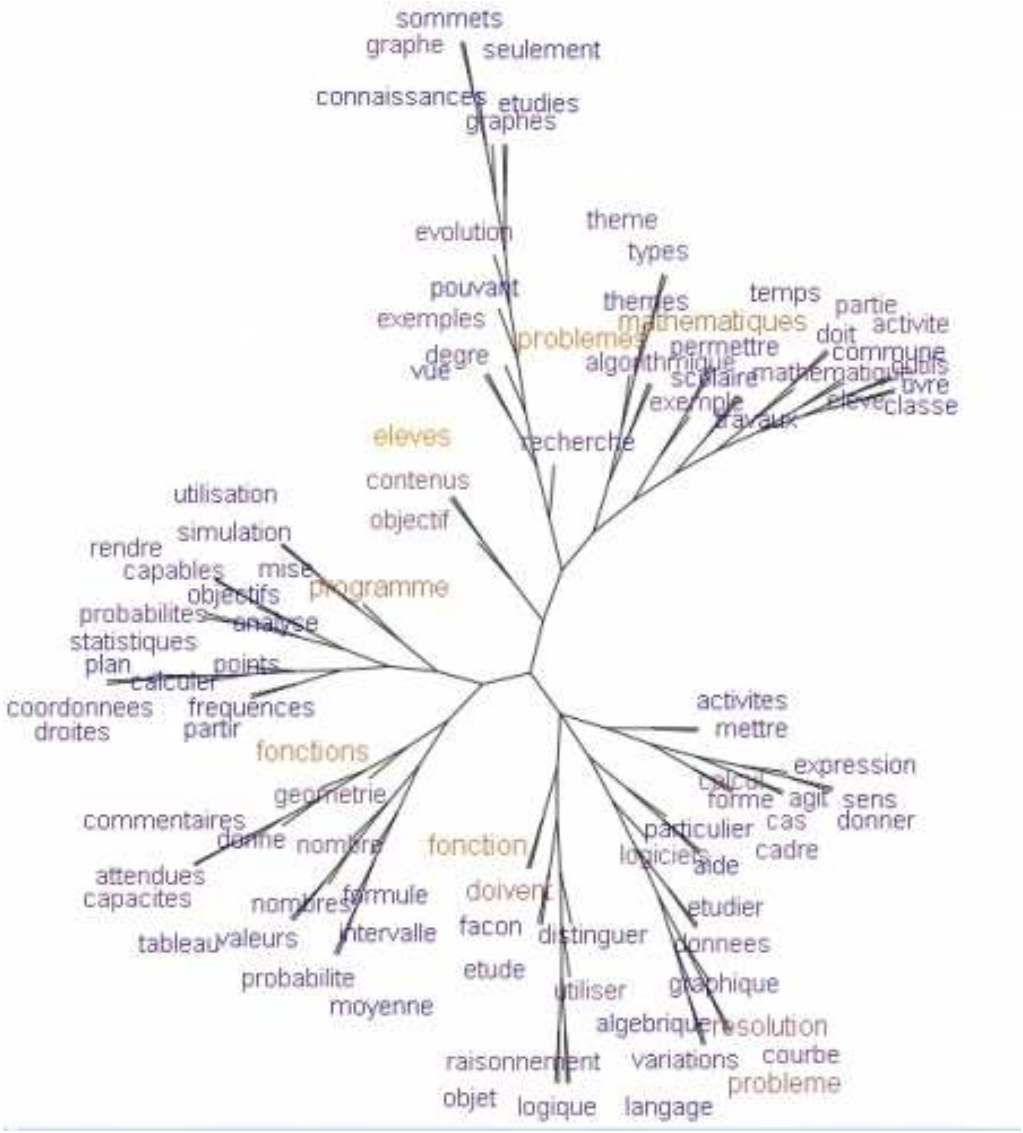


Il faut choisir un centre dans trois zones distinctes parmi les quatre.

Sous Scilab, on ne peut écrire que sur un segment. On choisira le paramètre angle dans l'instruction xstring.

Arrêt ?
On effectue autant d'itérations qu'il est possible de visualiser la construction de l'arbre.
On peut calculer la longueur d'une branche comme étant la limite des réductions pour des mots de taille 6 en moyenne, par exemple, puis choisir la dernière taille à considérer pour que l'on puisse atteindre 90% ou plus de la taille théorique de la branche ? (15 étapes ?)
SI 10 étapes, 3^10 = 59 049 mots !!

Le programme TreeCloud permet de construire de tels nuages arborés pour un texte quelconque. Les intérêts sont variés : visualisation rapide du contenu global d'un texte (rapport, livre...), analyse littéraire, comparaison de textes par comparaison de leurs nuages arborés.
Le texte du programme de mathématiques en seconde à la moulinette du Générateur d'arbres arborés, adresse obtenue à partir de la page TreeClouds.
Voilà les résultats en 100 mots :



Générateur d'arbres arborés : http://poulphunter1.free.fr/ArbreArbore/
Treeclouds : http://www.lirmm.fr/~gambette/ProgTreeCloud.php

ou bien

http://gambette.blogspot.com/2007/12/tag-cloud-tag-tree-nuage-arbor-1.html

Plusieurs méthodes peuvent être envisagées :

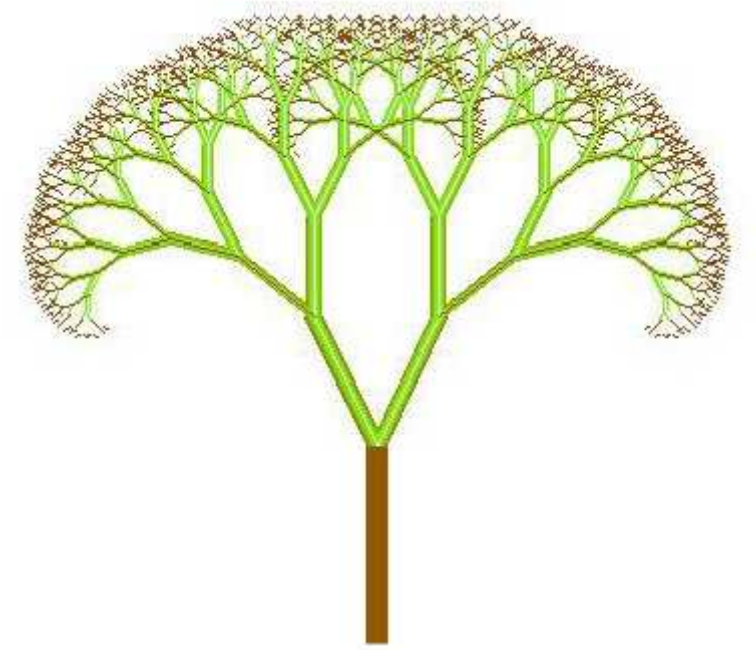
A partir de l'adresse : http://www.lirmm.fr/~gambette/treecloud/NuageArbore.cgi uniquement valide sur FireFox. On peut déposer son texte **complet** et l'arbre associé va apparaître.

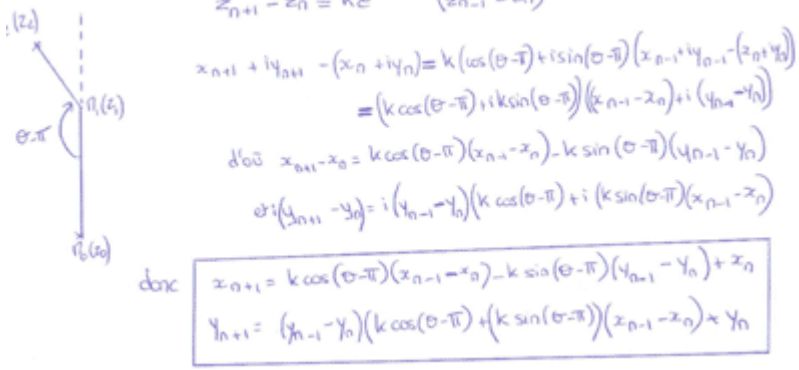
L'étape préalable suivante est bien plus intéressante :
Il s'agit d'extraire dans un texte les noms en excluant les déterminants et autres « petits mots ». A la main avec Scilab est tout à fait envisageable (il s'agit de mouliner une première fois en enlevant les , . ; : - … et autres caractères. Une deuxième fois en enlevant les majuscules et une troisième fois en enlevant les accents.
Construire un tableau qui prend chaque mot du texte.
Puis compter les occurrences de chacun de ces mots.

Une applet est disponible pour réaliser cela : http://www.freecorp.org/FRA/programmesdivers.htm
Dico est un petit utilitaire qui permet d'extraire le lexique des formes qui apparaissent dans un fichier texte.

Le tout pouvant se faire avec : http://www.splitstree.org/

Les arbres fractales





$$z_{n+1} - z_n = k e^{i(\theta-\pi)} (z_{n-1} - z_n)$$

$$x_{n+1} + iy_{n+1} - (x_n + iy_n) = k(\cos(\theta-\pi) + i\sin(\theta-\pi))(x_{n-1} + iy_{n-1} - (x_n + iy_n))$$

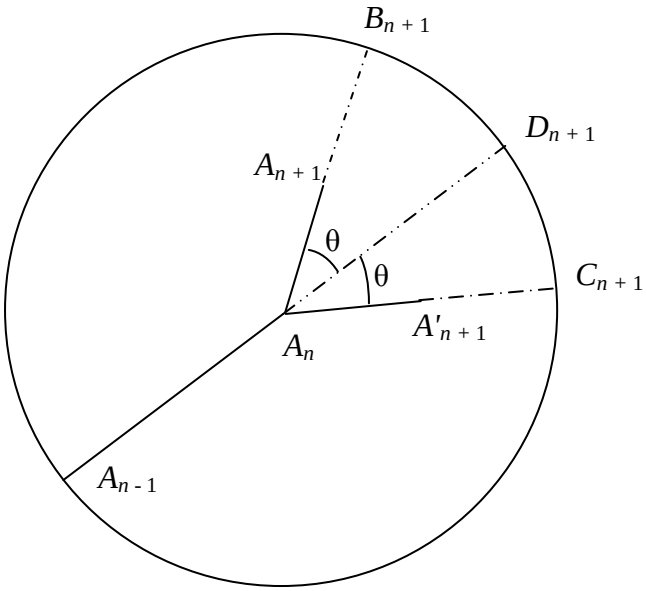
$$= (k\cos(\theta-\pi) + ik\sin(\theta-\pi))((x_{n-1} - x_n) + i(y_{n-1} - y_n))$$

d'où $x_{n+1} - x_n = k\cos(\theta-\pi)(x_{n-1} - x_n) - k\sin(\theta-\pi)(y_{n-1} - y_n)$

et $i(y_{n+1} - y_n) = i(y_{n-1} - y_n)(k\cos(\theta-\pi) + i(k\sin(\theta-\pi))(x_{n-1} - x_n)$

donc

$$\begin{cases} x_{n+1} = k\cos(\theta-\pi)(x_{n-1} - x_n) - k\sin(\theta-\pi)(y_{n-1} - y_n) + x_n \\ y_{n+1} = (y_{n-1} - y_n)(k\cos(\theta-\pi) + (k\sin(\theta-\pi))(x_{n-1} - x_n) + y_n \end{cases}$$



A partir des points A_{n-1} et A_n, on détermine les coordonnées de D_{n+1}
Puis celles de B_{n+1} et C_{n+1}
Et enfin celles de A'_{n+1} et A_{n+1}

Ces coordonnées seront stockées dans des tableaux de la forme (boucle en for)

$$t = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$t = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$t = \begin{bmatrix} \cdot & \cdot \\ \cdot & \cdot \end{bmatrix}$$

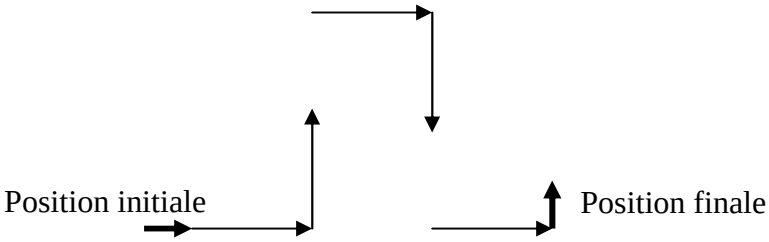
$$t = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix}$$

etc …

Il suffira de tracer les segments successifs sous scilab grâce à xsegs

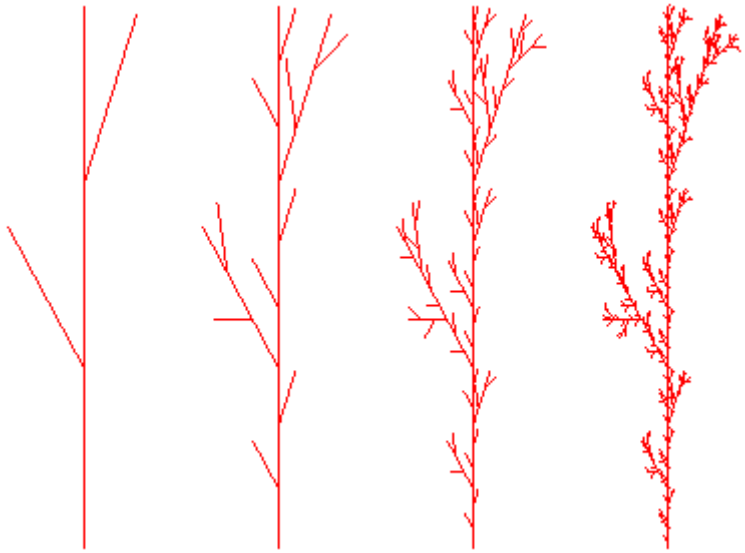
Les systèmes L pour la création de fractales

Soit une tortue qui obéit à quelques instructions simples :
F avance droit devant elle d'une longueur donnée en laissant une trace
f avance comme F mais ne laisse pas de trace
α tourne sur place d'un angle α dans le sens positif (inverse des aiguilles d'une montre)
-α tourne de -α (sens des aiguilles)
!' : évolution



F α F f -α F -α F f α F α

On peut utiliser les systèmes L pour dessiner des arbustes ou pour simuler la croissance des plantes. Mais il faut évidemment faire des branchements ; pour cela on utilise le symbole [pour commencer un embranchement, et] pour le terminer.
Soit par exemple la fractale :
Axiome : F
Règles de production :
F !' F [α F] F [-α F] F, les autres symboles ne changeant pas, et α = 30°.
Ci-dessus, le dessin obtenu après 1, 2, 3 et 4 itérations :



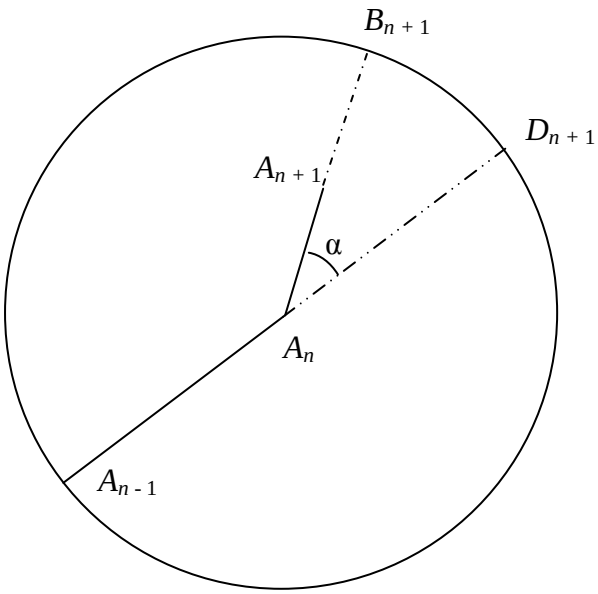
Le tout va se faire dans une chaîne de caractères :

C="F" on est à l'étape 1 dans le symbole initial est tracé avec une certaine longueur (dans l'exemple précédent, le trait est de longueur 10).

F est devenu C=" F [α F] F [-α F] F " avec α = 30°
A partir de la même position initiale, on trace le segment F dans un rapport d'un tiers puis on tourne de α et on continue de tracer F. On revient à l'extrémité du premier segment F puis on trace un second segment de longueur 1/3 puis on tourne de –α et on trace un segment F de longueur 1/3, on revient et on trace un dernier segment de longueur 1/3.

On peut transformer l'évolution " F [α F] F [-α F] F " en " F α F 180 F 180-α F -α F 180 F 180+α F "

Un rapport k suit la lecture de cette chaîne :
La lecture de α ou de –α transforme ce rapport *k* en 1/3 de k alors que la présence de 180-α ou 180+α transforme ce rapport en 3 fois ce rapport.



A partir des points A_{n-1} et A_n, on détermine les coordonnées de D_{n+1}
Puis celles de B_{n+1} enfin celles de A_{n+1}

Arbres fractales aléatoires

On peut utiliser les deux méthodes précédentes pour créer, grâce à l'algorithmique, des arbres fractales aléatoires.

Le problème étudié est celui de la croissance d'organismes végétaux en utilisant un *modèle symbolique* : on symbolise chaque élément nouveau de l'organisme par des lettres en indiquant par des règles de production la manière dont ils se transforment au bout d'une unité de temps.

1) Les règles suivantes expriment le développement annuel d'un tronc d'arbre :

M représente la moelle ; elle reste invariante	$M \rightarrow M$
B représente un anneau annuel ligneux	$B \rightarrow B$
A représente l'aubier ; c'est la partie active	$A \rightarrow BA$
E représente l'écorce ; elle reste invariante	$E \rightarrow E$

La configuration de départ est MAE et les configurations annuelles successives seront :

MBAE puis MBBAE puis MBBBAE

Ecrire la procédure **Developpement_Tronc;** qui permet d'afficher les développements successifs.
Vous demanderez au clavier le nombre de développements à afficher.
Vous pourrez considérer une variable de type string initialisée à MAE puis ferez évoluer cette variable à chaque développement.

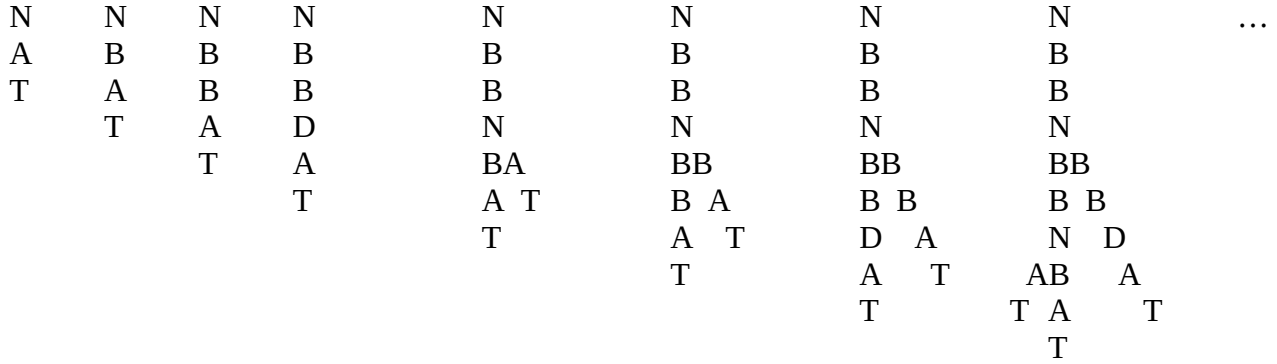
2) a) Construire la fonction **proba:byte;** qui, en utilisant un tirage aléatoire renvoie 1 avec une probabilité de $\frac{1}{3}$ et 2 avec une probabilité de $\frac{2}{3}$.

b)Les règles suivantes expriment le développement d'une algue ou d'une racine :

D représente le départ d'une racine	$D \rightarrow NAT$
B représente une partie inerte	$B \rightarrow B$
A représente une partie active	$A \rightarrow BA$ deux fois sur trois
	$A \rightarrow DA$ une fois sur trois
N représente un nœud	$N \rightarrow N$
T représente un bourgeon terminal	$T \rightarrow T$

La configuration de départ est D

Un exemple d'utilisation est le suivant dont on donne tout d'abord un aperçu graphique :



Mais nous utiliserons une représentation parenthésée (où chaque parenthèse représente le départ d'un nœud)

NAT puis NBAT puis NBBAT puis NBBDAT puis NBB(NAT)BAT puis NBB(NBAT)BBDAT puis NBB(NBBDAT)BB(NAT)BAT …

Ecrire la procédure **Developpement_Algue;** qui permet d'afficher les développements successifs.
Vous demanderez au clavier le nombre de développements à afficher.
Vous pourrez considérer une variable de type string initialisée à D.

(remarque : pour obtenir une description analogue pour le développement d'une feuille, avec trois nervures partant de chaque nœud, remplacer la règle $A \rightarrow DA$ par la règle $A \rightarrow DDD$)

Il s'agit de la construction précédente en L mais l'évolution se fait suivant un tirage aléatoire.

Voici un programme en Pascal

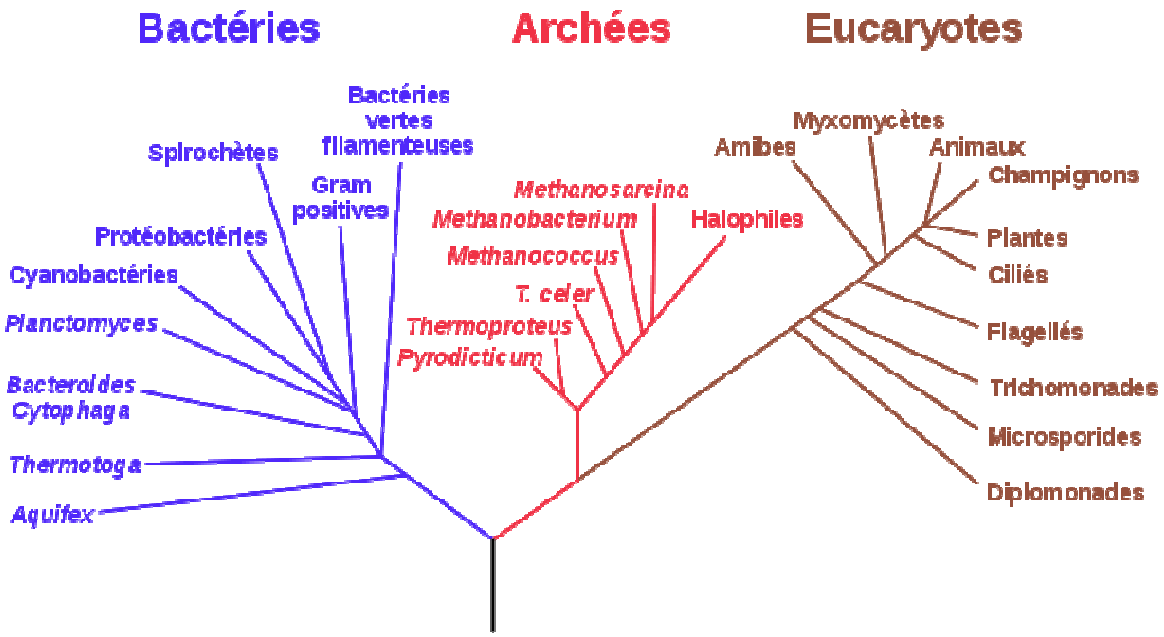
```
Program Modelisation;
Uses Crt;
Procedure Developpement_tronc;
var n,i:integer;
   s:string;
Begin
    clrscr;
    writeln('Entrez le nombre de d,veloppement d,sir,');readln(n);
    s:='MAE';writeln(s);
    for i:=1 to n do
       begin
           {seule la derniŠre position est … modifier}
           s[length(s)]:='A';s:=s+'E';writeln(s);
       end;
end;
Function proba:byte;
var x:byte;
begin
    x:=random(9);
    if x<3 then proba:=1
          else proba:=2;
end;
Procedure Developpement_Algue;
var n,i,j,p:integer;
   s,s1:string;
Begin
    clrscr;writeln('Entrez le nombre de d,veloppement d,sir,');readln(n);
    s:='NAT';writeln('D');writeln('NAT');
    for i:=1 to n do
       begin
           s1:='';randomize;
           for j:=1 to length(s) do
              case s[j] of
                 'D':s1:=s1+'(NAT)';'B':s1:=s1+'B';
                 'A':begin
                        p:=proba;
                        if p=1 then s1:=s1+'DA'
                              else s1:=s1+'BA';
                     end;
                 'N':s1:=s1+'N';'T':s1:=s1+'T';'(':s1:=s1+'(';')':s1:=s1+')';
              end;
           s:=s1;
           writeln(s);
       end;
end;
begin
    repeat developpement_algue;readln;until keypressed;
end.
```

Dans cette construction, on peut se contente du programme de construction, le reste peut se faire "à la main" sur une feuille.

J'ai utilisé les textes officiels sur les MPS pour créer la structure suivante. La taille des caractères est proportionnelle aux occurrences des mots dans le texte.

L'adresse : http://www.wordle.net/

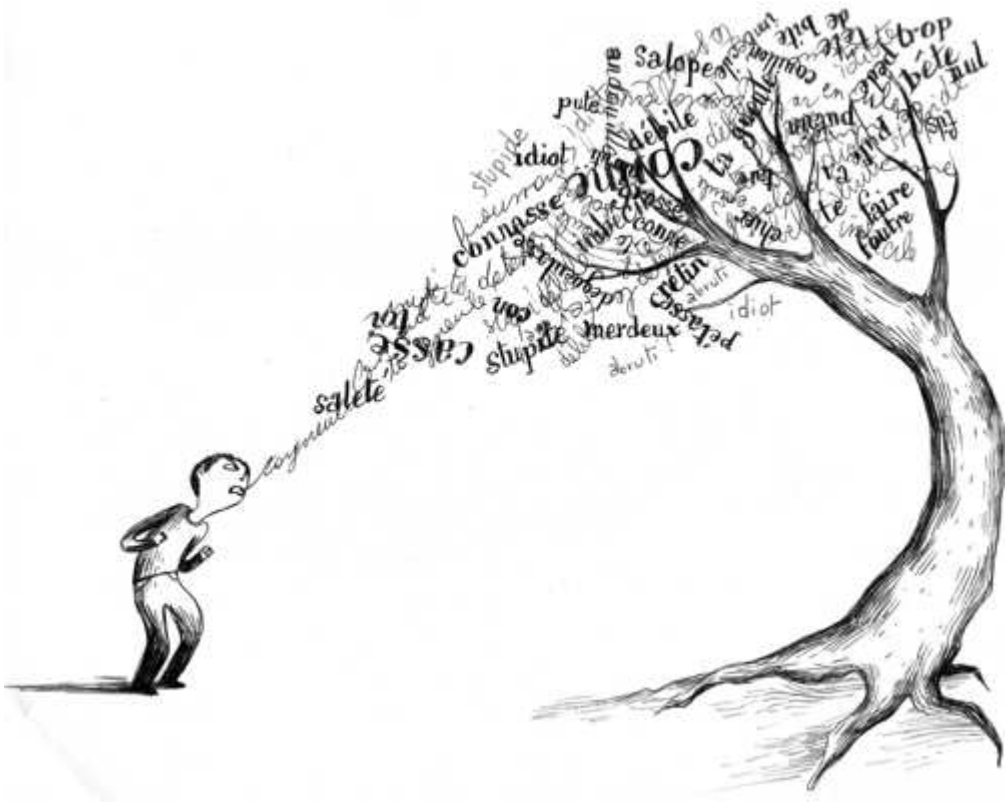
Arbre phylogénétique de la vie



Le dernier ancêtre commun universel est l'organisme primitif datant d'environ 3,6 à 4,1 milliards d'années et dont sont issues l'ensemble des espèces. L'acronyme LUCA venant de l'anglais Last Universal Common Ancestor est souvent utilisé pour désigner ce dernier ancêtre commun à toutes les formes de vie connues actuellement.

L'hypothèse courante est que tous les êtres vivants ont des ancêtres en commun, à une époque où la seule reproduction était la division cellulaire. Cela implique, l'existence dans le passé lointain d'une cellule telle que :

tous les êtres vivant actuellement en descendent ;
chacune de ses deux cellules filles a au moins un descendant vivant aujourd'hui (sans quoi une des cellules filles est en fait le LUCA réel).
LUCA ne doit pas être confondu avec le premier organisme vivant. Il est probable qu'il est lui-même issu d'une lignée évolutive et qu'il cohabitait avec d'autres formes de vie qui n'ont pas laissé de descendants.



Mathieu Salade m'a raconté que quand il était petit, il connaissait un petit garçon très grossier. Sa mère lui dit un jour : si tu veux dire des grossièretés, défoule-toi sur cet arbre, tu as le droit de lui dire tout ce que tu veux, tu ne sera pas grondé. Ce jour là, l'arbre est devenu l'arbre à gros mots.
Je vous laisse une semaine (je pars dans un pays pas très lointain!) à bientôt!

***l'arbre à gros mots*, http://www.princessecamcam.fr/**

L’arbre généalogique

a) Je veux faire mon arbre généalogique ascendant, c’est à dire celui qui contient mes deux parents (première génération), mes quatre
grands-parents (deuxième génération)... jusqu’à mes ancêtres de la cinquième génération.
Combien cet arbre comporte-t-il de personnes en me comptant ?

b) On numérote ainsi les membres de l’arbre : je porte le numéro 1, mon père le numéro 2, ma mère le numéro 3 et les parents du
numéro n sont numérotés 2n pour le père et 2n + 1 pour la mère ; ainsi, le numéro 4 est le père de mon père.
Sur le même modèle et en utilisant uniquement les mots père, mère, de, la, du, mon, préciser qui porte le numéro 37.