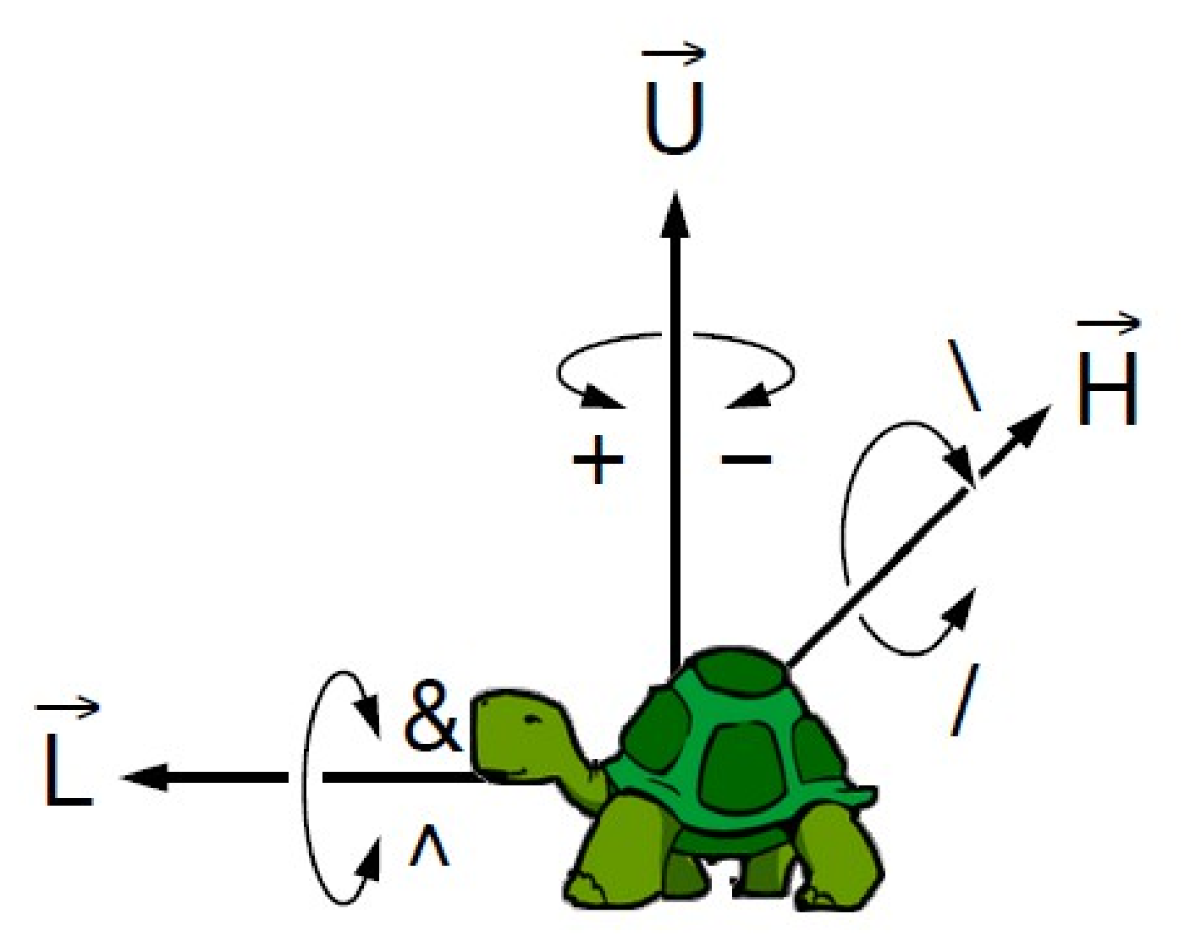
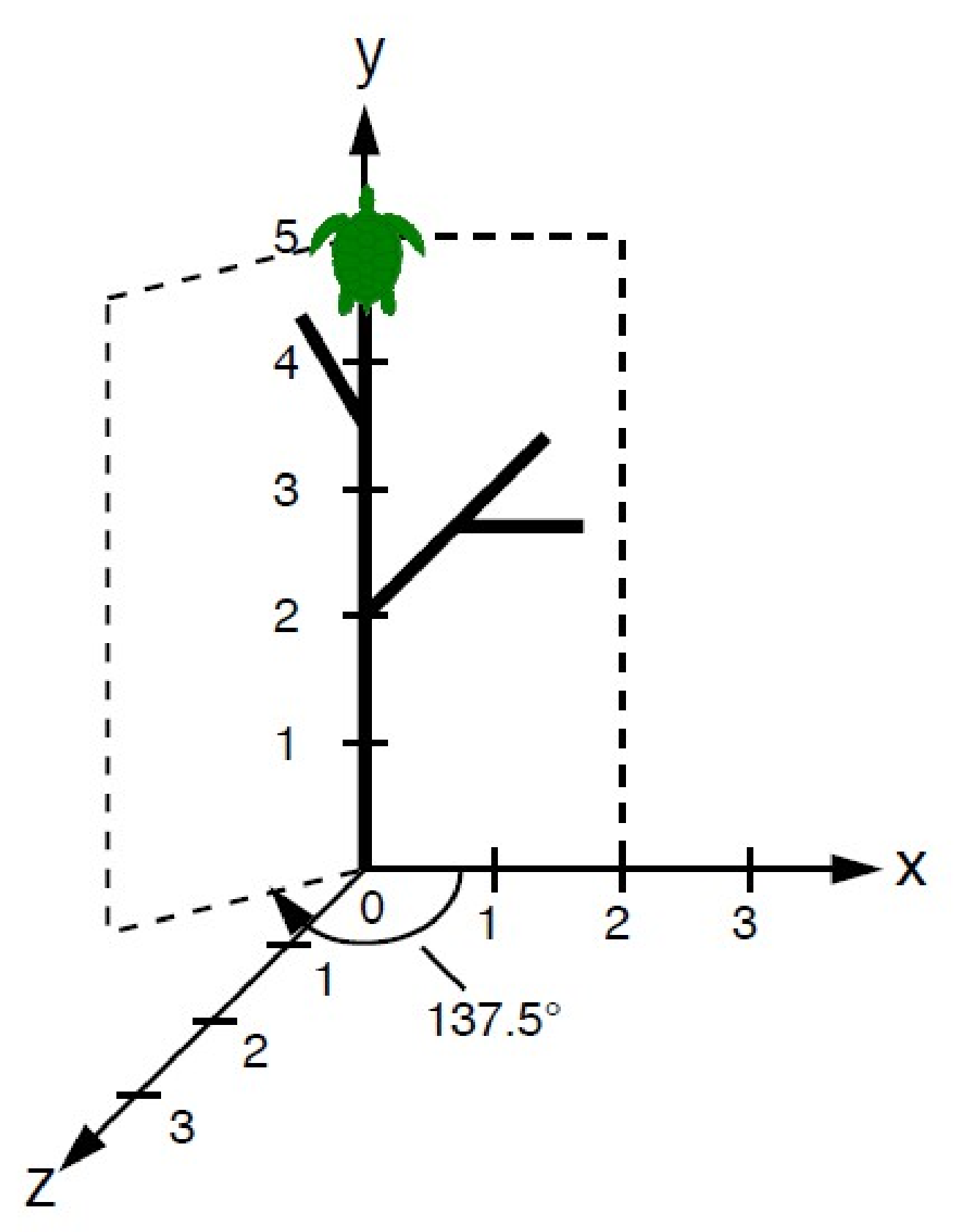
Procedural generation of 3D trees based on L-Systems in real-time graphics

Teacher
Jakob Andreas Bærentzen

Nicolas Barthelemy
MSc exchange student at DTU

Teacher Asistant
Alessandro Dal Corso





Principle : Turtle graphics

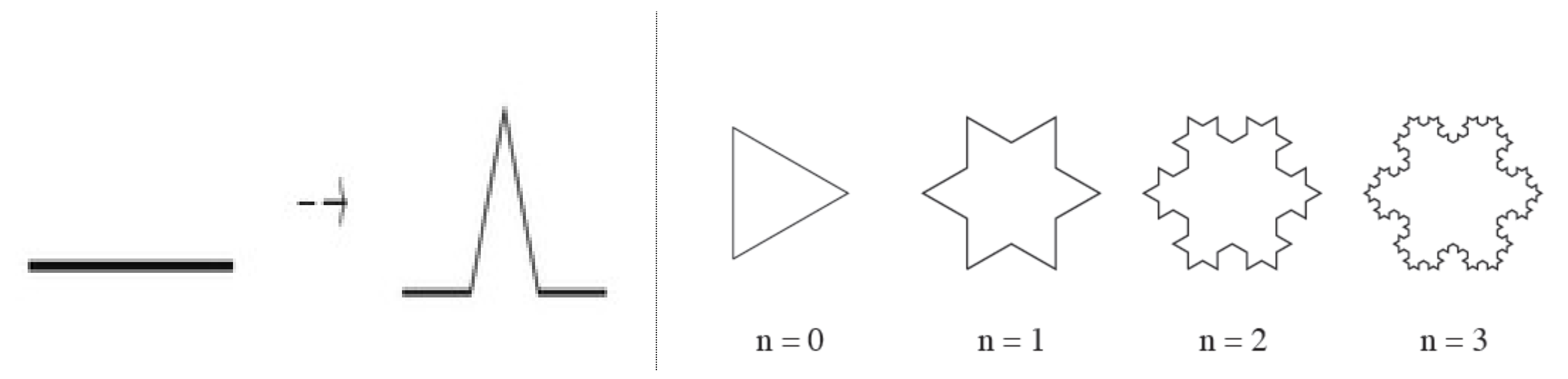
- The turtle moves and rotate according to its orientation thanks to basic instructions like :
  FORWARD 10
  TURN 90
- When it moves it draws a line, we then obtain a sketch (cf right picture).
- We apply this principle in 3D by creating cones to draw our trees. This is called **Grammar Based Modelling**.

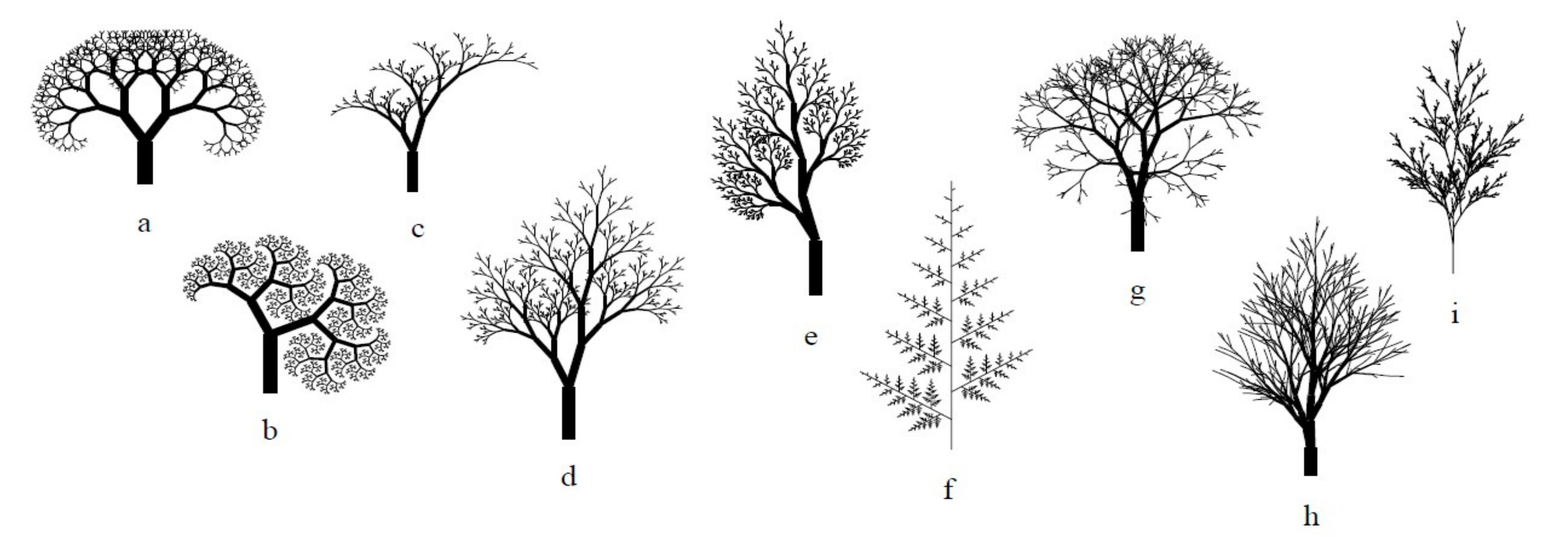
Turtle rules

- The pictures are an example to understand how **Grammar Based Modelling** work.
- We replace it by the expression after the arrow : we move only one third of the size of the line « s/3 », « +(60) » stands for turn 60 degrees and then move again of « s/3 », etc.
- This way we can get the snowflakes on the right bottom image, by applying several times this simple rule.

Principle example : Koch's snowflake

$$F(s) \rightarrow F(s/3) + (60)\ F(s/3) - (120)\ F(s/3) + (60)\ F(s/3)$$





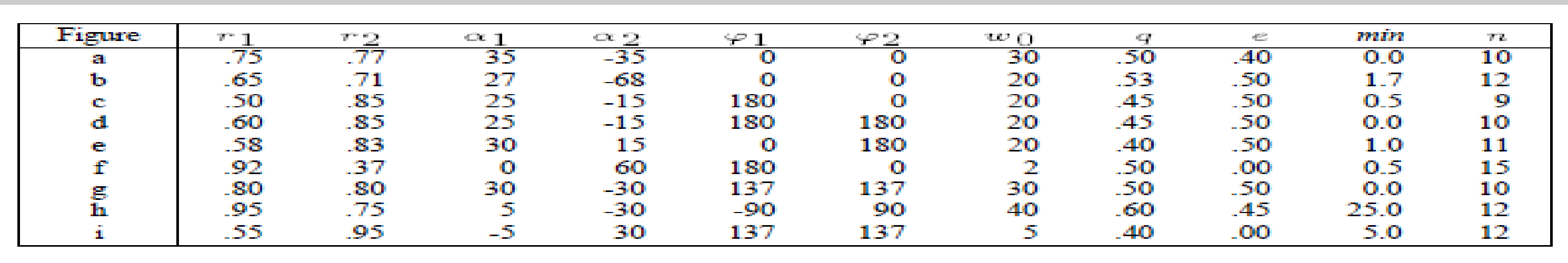


Figure	r_1	r_2	α_1	α_2	φ_1	φ_2	w_0	q	e	min	n
a	.75	.77	35	-35	0	0	30	.50	.40	0.0	10
b	.65	.71	27	-68	0	0	20	.53	.50	1.7	12
c	.50	.85	25	-15	180	0	20	.45	.50	0.5	9
d	.60	.85	25	-15	180	180	20	.45	.50	0.0	10
e	.58	.83	30	15	0	180	20	.40	.50	1.0	11
f	.92	.37	0	60	180	0	2	.50	.00	0.5	15
g	.80	.80	30	-30	137	137	30	.50	.50	0.0	10
h	.95	.75	5	-30	-90	90	40	.60	.45	25.0	12
i	.55	.95	-5	30	137	137	5	.40	.00	5.0	12

Theory to visual models

- Source : L-systems: from the Theory to Visual Models of Plants by Przemyslaw Prusinkiewicz, Jim Hanan, Mark Hammel and Radomir Mech.

In our case we use a more complicated rule : Prusinkiewicz's rule.

$$\begin{aligned}
\omega &: \quad A(100, w_0) \\
p_1 &: \quad A(s, w) \quad : s >= min \quad \rightarrow \quad !(w)F(s) \\
&\qquad [+(\alpha_1)/(\varphi_1)A(s * r_1, w * q \wedge e)] \\
&\qquad [+(\alpha_2)/(\varphi_2)A(s * r_2, w * (1-q) \wedge e)]
\end{aligned}$$

This one enables for instance our turtle to go back (so we get the branches of the tree) or to change the thickness of the branch.
On the left pictures are the results we get with this method.

- Top image : different trees obtainable
- Bottom image : corresponding values for the rule

Tools and implementation

- We use C++ and OpenGL 3.2 in Qt Creator
- In the course we already had an implementation with a trivial production rule. The goal was to implement Prusinkiewicz's rule instead and implement the turtle functions in a prepared class.
- Then I ran it and got the following results :
  - On the left picture I used the values above
  - On the right picture I used my custom values