

1. Starting with the letter b (step 1), consider the rewriting rules

$$b \rightarrow a; \quad a \rightarrow ab.$$

For example, the first few 'words' formed by the above rewriting rules are

$$b, a, ab, aba, abaab, abaababa, \dots$$

Let $C(n)$ be the number of letters in the n -th word (n is a positive integer). For instances,

$$C(1) = 1, \quad C(2) = 1, \quad C(3) = 2, \quad C(4) = 3, \quad C(5) = 5, \quad C(6) = 8, \dots$$

Prove that

$$C(n + 1) = C(n) + C(n - 1)$$

for *all* integers $n \geq 2$. (*Caution*: Math. induction may *not* work well here.)

2. Consider the turtle's movement:

F means advancing 1 unit in the direction of the turtle's head;

$+/-$ means, respectively, turning the turtle's head to the right/left by d degrees - the position of the turtle remains unchanged.

Take $d = 90^\circ$ and start with $F + F + F + F$ (0 iteration). In the subsequent steps, let the rewriting rule be

$$F \mapsto F - F + F.$$

(a) Sketch the first two iterations.

(b) It is observed that in the first few iterations, the turtle always returns to the same position where it starts. Does this hold for *all* iterations? Present the key ideas critically and carefully to support your answers. **(Q.3 in the next page.)**

3. In a simple Lindenmayer system,

[] means performing the procedure(s) inside the square brackets and then returning to the (starting) state just 'behind' the left bracket;

{ } means making all line segments inside the brackets twice as long.

Originally the turtle is located at $(0, 0)$ on the plane with its head 'up' (that is, facing the positive y -direction). Taking $d = 20^\circ$, obtain the graphs of the first 5 iterations with the rewriting rules:

(I) starting with F (iteration 0);

(II) in each step, rewriting F by $\{F\} [+F] [-F]$.

(For the last two graphs, you may resort to sketching and rescaling.) In each iteration, where is the turtle last position?

For the sake of understanding how [] and { } clarify the presentation, represent the second iteration as a turtle graph *without* using [] and { }. How about the third iteration? Consider also whether it is possible to reformulate rewriting rule (II) *without* using [] and { }.