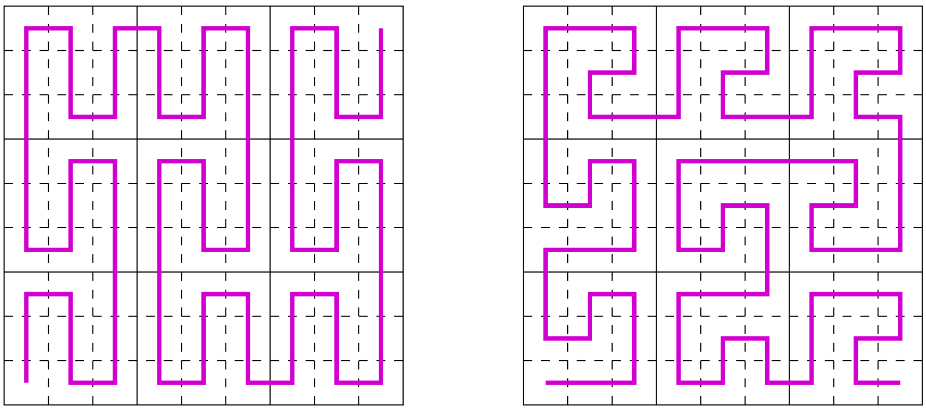
Technische Universität München
Lehrstuhl für Informatik V
Univ.-Prof. Dr. M. Bader
M.Sc. A. Mo-Hellenbrand, M.Sc. A. Schwarz

SS 2016
Worksheet 10
22.6.2016

Algorithms of Scientific Computing

Exercise 1: Grammars for Peano curves



(a) Switch-back Peano curve (b) Peano-Meander curve

Figure 1: Two distinct Peano-type space filling curves

a) The Peano curve from Figure 1(a) can be constructed by the following grammar:

- non-terminal symbols: $\{P, Q, R, S\}$, start symbol P
- terminal symbols: $\{\uparrow, \downarrow, \leftarrow, \rightarrow\}$
- production rules:

$$
\begin{aligned}
P &\longleftarrow P \uparrow Q \uparrow P \rightarrow S \downarrow R \downarrow S \rightarrow P \uparrow Q \uparrow P \\
Q &\longleftarrow Q \uparrow P \uparrow Q \leftarrow R \downarrow S \downarrow R \leftarrow Q \uparrow P \uparrow Q \\
R &\longleftarrow R \downarrow S \downarrow R \leftarrow Q \uparrow P \uparrow Q \leftarrow R \downarrow S \downarrow R \\
S &\longleftarrow S \downarrow R \downarrow S \rightarrow P \uparrow Q \uparrow P \rightarrow S \downarrow R \downarrow S
\end{aligned}
$$

The derivation of the grammar from the construction principle of the curve is shown in Figure 2. Note that the patterns P and R as well as the patterns Q and S only differ with respect to the traversal direction.

For the Peano-meander curve given in Figure 1(b) we get the following grammar:

- non-terminal symbols: $\{M, N, L, W\}$, start symbol N
- terminal symbols: $\{\uparrow, \downarrow, \leftarrow, \rightarrow\}$

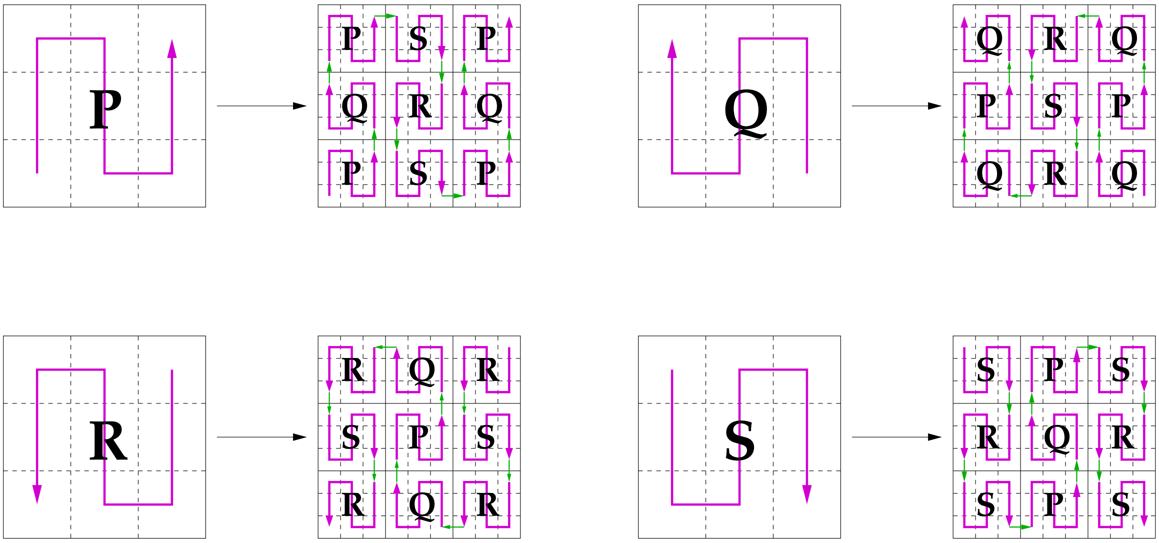


Figure 2: Relation between grammar and construction of the Peano curve

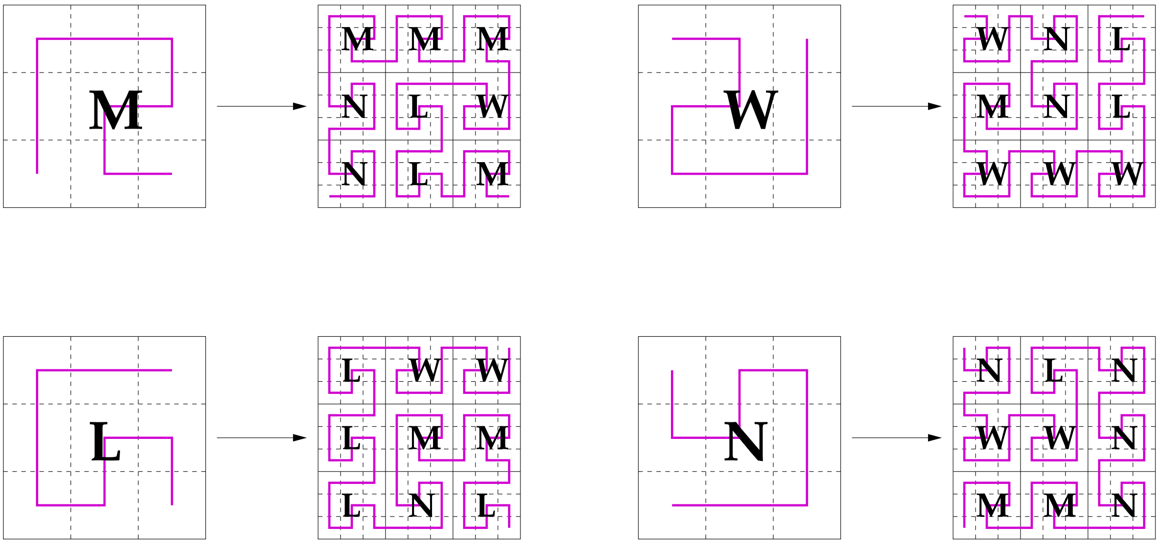


Figure 3: Relation between grammar and construction of the Peano curve

– production rules:

$$\begin{aligned}
M &\longleftarrow N \uparrow N \uparrow M \rightarrow M \rightarrow M \downarrow W \leftarrow L \downarrow L \rightarrow M \\
N &\longleftarrow M \rightarrow M \rightarrow N \uparrow N \uparrow N \leftarrow L \downarrow W \leftarrow W \uparrow N \\
L &\longleftarrow W \leftarrow W \leftarrow L \downarrow L \downarrow L \rightarrow N \uparrow M \rightarrow M \downarrow L \\
W &\longleftarrow L \downarrow L \downarrow W \leftarrow W \leftarrow W \uparrow M \rightarrow N \uparrow N \leftarrow W
\end{aligned}$$

The relation between grammar and construction is shown in Figure 3. This time we don't need the traversal direction to distinguish between the four patterns.

b) See the attached Python program.

Exercise 2: Real Turtle for the Hilbert Curve

The grammar from the lecture can be directly converted to a "real Turtle graphics"-grammar: We use the same symbols but change the global movement commands into local rotation and movement commands. For this approach the turtle needs to be in a specific orientation at the beginning and the end of the steps of a pattern. So, we can't avoid that it turns several times on the same spot before doing the next step.

We can use the following approach to solve this problem: We distinguish the cases by the edge over which the turtle exits the subsquare. It enters the subsquare always over the same edge, since we only consider the local subproblem, where the turtle needs to traverse the current subsquare, independent of the actual global orientation of this subsquare. (see Figure 4)

Basically we only get three cases, since we can leave the subsquare through the edge straight ahead, through the one to the left or through the one to the right. However, each of these cases is further separated into two cases, since it is important if we enter the subsquare on the left-hand or on the right-hand side. This gives us six cases in total, which are shown in Figure 4.

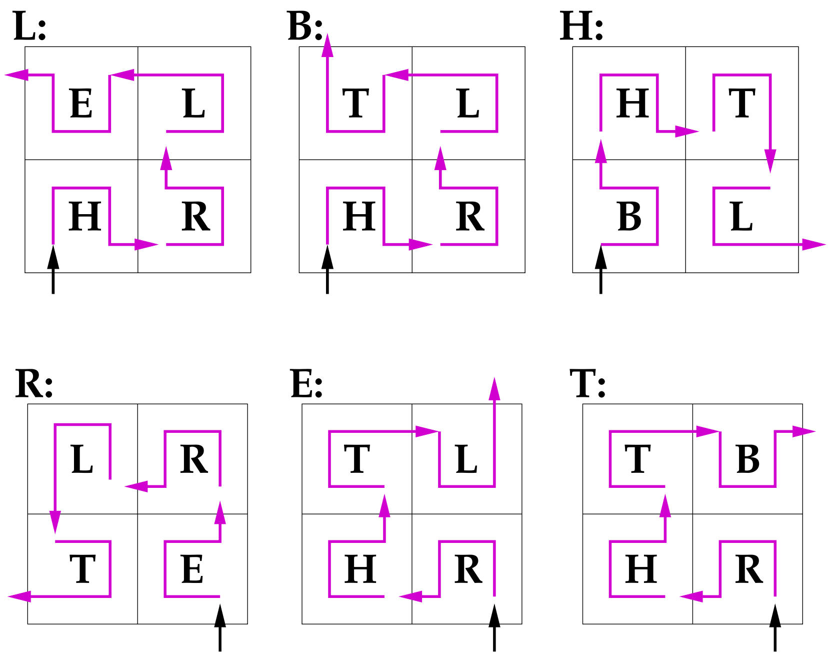


Figure 4: Relation between grammar and construction of the Hilbert curve

Let us define now the symbols for the grammar:

- Non-terminals: $\{H, L, B, E, R, T\}$, start symbol H
- Terminals: {⬆, ↰, ↱}

The signs represent the actions "one step ahead", "turn to the left" and "turn to the right".

For finding the productions we need to map the nonterminal symbols appropriately to the patterns.

Version 1

In Version 1 the patterns include always the step to the next subsquare. So, we get the following productions:

H	$\longleftarrow$	$B\,H\,T\,L$		H	$\longleftarrow$	↑↱↑↱↑↰↑
B	$\longleftarrow$	$H\,R\,L\,T$		B	$\longleftarrow$	↱↑↰↑↰↑↱↑
L	$\longleftarrow$	$H\,R\,L\,E$		L	$\longleftarrow$	↱↑↰↑↰↑↑
E	$\longleftarrow$	$R\,H\,T\,L$		E	$\longleftarrow$	↰↑↱↑↱↑↰↑
T	$\longleftarrow$	$R\,H\,T\,B$		T	$\longleftarrow$	↰↑↱↑↱↑↑
R	$\longleftarrow$	$E\,R\,L\,T$		R	$\longleftarrow$	↑↰↑↰↑↱↑

As usual we need to constrict the production rules: All non-terminals have to be replaced simulatenously. Moreover, we always use either the left (recursive decent) or the right (recursion abortion) production rules.

Version 2

In version 1 we notice that the last step of a terminal production is always a *forward* command, which is the step towards the next subsquare. We can extract this step from the terminal productions and insert it into the non-terminal productions:

H	$\longleftarrow$	B ↑ H ↑ T ↑ L		H	$\longleftarrow$	↑↱↑↱↑↰
B	$\longleftarrow$	H ↑ R ↑ L ↑ T		B	$\longleftarrow$	↱↑↰↑↰↑↱
L	$\longleftarrow$	H ↑ R ↑ L ↑ E		L	$\longleftarrow$	↱↑↰↑↰↑
E	$\longleftarrow$	R ↑ H ↑ T ↑ L		E	$\longleftarrow$	↰↑↱↑↱↑↰
T	$\longleftarrow$	R ↑ H ↑ T ↑ B		T	$\longleftarrow$	↰↑↱↑↱↑
R	$\longleftarrow$	E ↑ R ↑ L ↑ T		R	$\longleftarrow$	↑↰↑↰↑↱

Here the same constrictions for the production rules as in version 1 hold. From this productions you can easily see that two rotations may never appear consecutively, since the terminal productions do not contain consecutive rotations and the non-terminal productions are always separated by forward steps.

Version 3

The non-terminal productions of version 2 show a similar structure as the basic patterns from the beginning. I.e. if we drop the non-terminal symbols we get almost the atomic traversal pattern (i.e. ″north″, ″south″, ″west″, ″east″), but only described by forward steps. If we try to map the needed rotations to the non-terminal symbols, we see that H and T must describe a rotation to the right, while L and R describe a rotation to the left.

By this we can retrieve the following productions (with ϵ-productions) for B and E):

$$
\begin{array}{lcl@{\qquad\qquad}lcl}
H & \longleftarrow & B \Uparrow H \Uparrow T \Uparrow L & H & \longleftarrow & \Rsh \\
B & \longleftarrow & H \Uparrow R \Uparrow L \Uparrow T & B & \longleftarrow & \epsilon \\
L & \longleftarrow & H \Uparrow R \Uparrow L \Uparrow E & L & \longleftarrow & \Lsh \\
E & \longleftarrow & R \Uparrow H \Uparrow T \Uparrow L & E & \longleftarrow & \epsilon \\
T & \longleftarrow & R \Uparrow H \Uparrow T \Uparrow B & T & \longleftarrow & \Rsh \\
R & \longleftarrow & E \Uparrow R \Uparrow L \Uparrow T & R & \longleftarrow & \Lsh
\end{array}
$$

Again the same constrictions hold as in the versions 1 and 2.

The resulting Python program is attached.