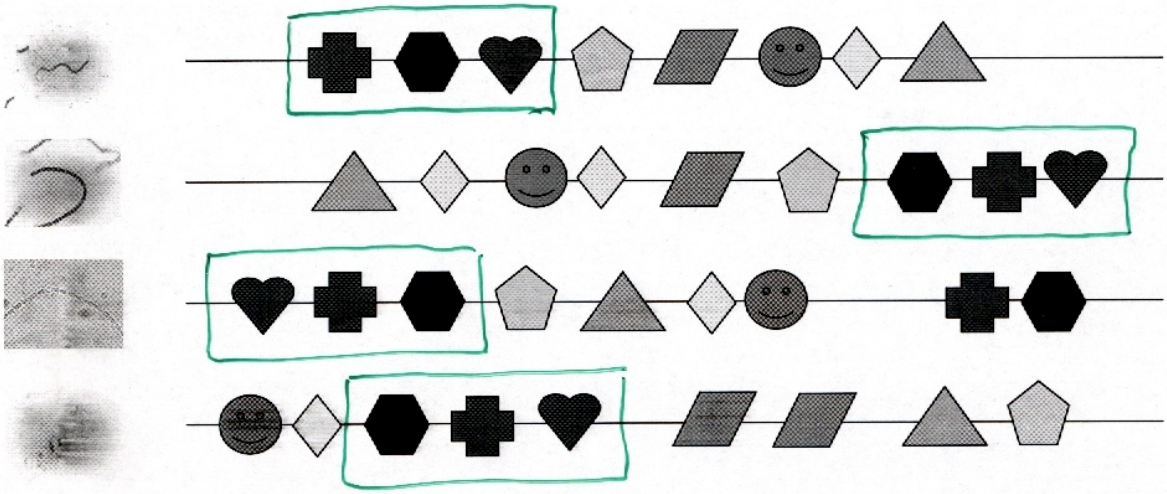
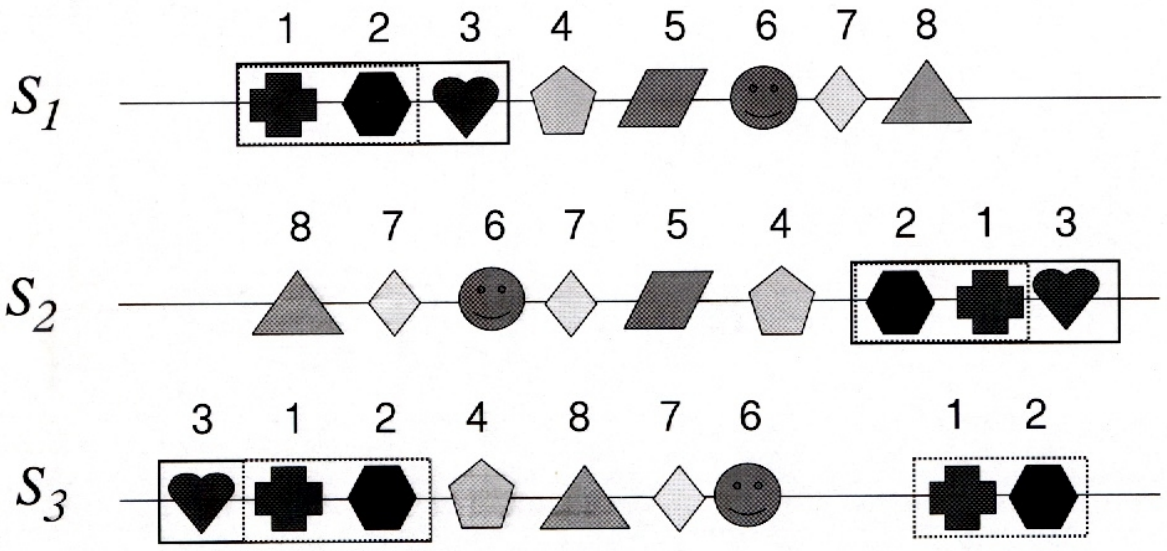
Bakteriengenome



Genome als Strings von Zahlen



Arrays POS und NUM

S_1 = 3 1 2 3 1 5 2 6

POS[1] = 2,5
POS[2] = 3,7
POS[3] = 1,4
POS[4] = empty
POS[5] = 6
POS[6] = 8

NUM[*i*,*j*] :

i^j	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*
1	1	2	3	3	3	4	4	5
2		1	2	3	3	4	4	5
3			1	2	3	4	4	5
4				1	2	3	4	5
5					1	2	3	4
6						1	2	3
7							1	2
8								1

Algorithmus Connecting Intervals

1: **pre-processing:** build data structures POS and NUM
2: **for** $i = 1, \ldots, |S_2|$ **do**
3: $OCC[c] \leftarrow 0$ for each character c in Σ, $|OCC| \leftarrow 0$
4: $j \leftarrow i$
5: **while** $j \leq |S_2|$ **and** (i, j) is left-maximal in S_2 **do**
6: $c \leftarrow S_2[j]$
7: $OCC[c] \leftarrow 1$, update $|OCC|$ if necessary
8: **while** (i, j) is not right-maximal in S_2 **do**
9: $j \leftarrow j + 1$
10: **end while**
11: **for** each position p in $POS[c]$ **do**
12: mark position p in S_1
13: find the maximal interval $(start, end)$ of positions marked so far that contains position p → merge p with left/right maximal interval
14: **if** $NUM(start, end) = |OCC|$ **and** $(start, end)$ is maximal in S_1 **then**
15: output the pair $((i, j), (start, end))$
16: **end if**
17: **end for**
18: $j \leftarrow j + 1$
19: **end while**
20: **end for**

$S_2[i-1] \neq S_2[j]$

$OCC[S_2[j+1]] = 0$

Beispiel für den Algorithmus