Algorithm 1 Check if the a motif automaton recognizes only the prefix of size $m-1$ of a motif P of size m

---

1: **function** SEARCHMOTIFLASTPREFIX(S: Array(n), A: $\langle S, s_0, T, \Sigma, f \rangle$, P: Array(m))
2:     **returns** A set of positions where the motif has been found
3:     $s \leftarrow s_0$
4:     $i \leftarrow 0$
5:     $T_{new} \leftarrow \{\}$
6:     **for** $s \in S$ **do**
7:         **for** $a \in \Sigma$ **do**
8:             **for** $t \in T$ **do**
9:                 **if** $\exists f(s, a)$ and $f(s, a) = t$ **then**
10:                     $T_{new} \leftarrow T_{new} \cup s$
11:                 **end if**
12:             **end for**
13:         **end for**
14:     **end for**
15:     **while** $i < n$ **do**
16:         **if** $s \in T_{new}$ **then**
17:             **return** true
18:         **end if**
19:         $s \leftarrow f(s, S[i])$
20:         $i++$
21:     **end while**
22:     **return** false
23: **end function**

Algorithm 2 Check if the a motif automaton recognizes only the prefix of size $m-1$ of a motif P of size m, knowing the sequence of the motif

1: **function** SEARCHMOTIFLASTPREFIX(S: Array(n), A: $\langle S, s_0, T, \Sigma, f \rangle$, P: Array(m))
2:     **returns** A set of positions where the motif has been found
3:     $s \leftarrow s_0$
4:     $i \leftarrow 0$
5:     **while** $i < n$ and $f(s, P[m-1]) \notin T$ **do**
6:         $s \leftarrow f(s, S[i])$
7:         $i++$
8:     **end while**
9:     **if** $f(s, P[m-1]) \in T$ **then**
10:         **return** true
11:     **else**
12:         **return** false
13:     **end if**
14: **end function**