

Automaten und Berechenbarkeit

11. Übungsserie

Aufgabe 1:

	a	b	c	x	\square
q_a	(q_b, x, R)	(q_T, b, N)	(q_T, c, N)	(q_a, x, R)	(q_F, \square, L)
q_b	(q_b, a, R)	(q_c, x, R)	(q_T, c, N)	(q_b, x, R)	(q_T, \square, L)
q_c	(q_T, a, N)	(q_c, b, R)	(q_L, x, L)	(q_c, x, R)	(q_T, \square, L)
q_L	(q_L, a, L)	(q_L, b, L)	(q_L, c, L)	(q_L, x, L)	(q_a, \square, R)
q_T	(q_T, a, R)	(q_T, b, R)	(q_T, c, R)	(q_T, x, R)	(q_{T_2}, \square, L)
q_{T_2}	(q_{T_2}, \square, L)	(q_{T_2}, \square, L)	(q_{T_2}, \square, L)	(q_{T_2}, \square, L)	(q_{T_2}, \square, N)
q_F	—	—	—	(q_F, \square, L)	(q_F, \square, N)

In q_a sucht die Turingmaschine nach einem a . Bei Fund wird das a weggestrichen und die Turingmaschine sucht nach einem b , usw. Wenn kein a gefunden wird, dann war $w \in L_3$ und es wird aufgeräumt (TM kommt in den Finalzustand). Wenn nach einem anderen Buchstaben als a gesucht wird, der jedoch nicht gefunden wird, dann ist die Anzahl der Buchstaben nicht gleich und es wird aufgeräumt, wobei die Turingmaschine nicht in den Finalzustand kommt.

Da die TM in jedem Durchlauf (fast) einmal von links nach rechts und zurück läuft, ist $t(n) \in O(n^2)$.

Aufgabe 2:

$|Q| \cdot |\Gamma|^{s(n)} \cdot s(n)$ ist die maximale Anzahl von Konfigurationen für eine Turingmaschine, die $s(n)$ Zellen benutzt. Wird eine Konfiguration von einer Turingmaschine ein zweites Mal berechnet, so ist sie eine Nachfolgekonfiguration von sich selbst und die Turingmaschine terminiert nicht. Da also eine Turingmaschine jede Konfiguration nur maximal einmal berechnen kann, wenn sie terminiert, braucht sie maximal $|Q| \cdot |\Gamma|^{s(n)} \cdot s(n)$ Takte.

Aufgabe 3:

(a)	a	b	A	B	$\#$	\square
q_0	—	—	—	—	—	$(q_1, \#, R)$
q_1	—	—	—	—	—	(q_{copy}, a, R)
q_{copy}	(q_{copy}, a, R)	$q_{copy, b, R}$	—	—	—	$(q_{cp_l}, \#, L)$
q_{cp_r}	(q_{cp_r}, a, L)	(q_{cp_r}, b, L)	—	—	$(q_{cp_l}, \#, L)$	—
q_{cp_l}	(q_{cp_l}, a, L)	(q_{cp_l}, b, L)	(q_{cp_1}, A, R)	(q_{cp_1}, B, R)	$(q_{cp_1}, \#, R)$	—
q_{cp_1}	(q_{paste_a}, A, R)	(q_{paste_b}, B, R)	—	—	$(q_{clean}, \#, L)$	—
q_{paste_a}	(q_{paste_a}, a, R)	(q_{paste_a}, b, R)	—	—	$(q_{paste_a}, \#, R)$	(q_{cp_r}, a, L)
q_{paste_b}	(q_{paste_b}, a, R)	(q_{paste_b}, b, R)	—	—	$(q_{paste_b}, \#, R)$	(q_{cp_r}, b, L)
q_{clean}	(q_{clean}, a, R)	(q_{clean}, b, R)	(q_{clean}, a, L)	(q_{clean}, b, L)	$(q_{clean}, \#, R)$	(q_{inc}, \square, L)
q_{inc}	(q_{copy}, b, L)	(q_{inc}, a, L)	—	—	$(q_{grow}, \#, R)$	—
q_{grow}	(q_{grow}, a, L)	—	—	—	—	(q_{copy}, a, R)

Erst werden λ und a erzeugt. Dann kopiert die TM das vorherige Wort und "addiert" 1 (wiederholt). Wenn alle möglichen Wörter der gleichen Länge geschrieben wurden, besteht das letzte Wort nur aus b und die Inkrementierung erweitert (neben der Invertierung aller vorherigen b) das Wort um ein a .

(b)

	a	A	$\#$	\square
q_0	—	—	—	$(q_1, \#, R)$
q_1	—	—	—	(q_{copy}, a, R)
q_{copy}	(q_{copy}, a, R)	—	—	$(q_{cp_l}, \#, L)$
q_{cp_r}	(q_{cp_r}, a, L)	—	$(q_{cp_l}, \#, L)$	—
q_{cp_l}	(q_{cp_l}, a, L)	(q_{cp_1}, A, R)	$(q_{cp_1}, \#, R)$	$(q_{cp_1}, \#, R)$
q_{cp_1}	(q_{paste_a}, A, R)	—	$(q_{clean}, \#, L)$	—
q_{paste_a}	(q_{paste_a}, a, R)	—	$(q_{paste_a}, \#, R)$	(q_{cp_r}, a, L)
q_{clean}	$(q_{clean}, a, R)(q_{clean}, a, L)$	$(q_{clean}, \#, R)$	(q_{copy_2}, \square, L)	—
q_{copy_2}	(q_{copy_2}, a, R)	—	—	$(q_{cp_{l_2}}, \#, L)$
$q_{cp_{r_2}}$	$(q_{cp_{r_2}}, a, L)$	—	$(q_{cp_{l_2}}, \#, L)$	—
$q_{cp_{l_2}}$	$(q_{cp_{l_2}}, a, L)$	$(q_{cp_{1_2}}, A, R)$	$(q_{cp_{1_2}}, \#, R)$	—
$q_{cp_{1_2}}$	$(q_{paste_{a_2}}, A, R)$	—	$(q_{clean_2}, \#, L)$	—
$q_{paste_{a_2}}$	$(q_{paste_{a_2}}, a, R)$	—	$(q_{paste_{a_2}}, \#, R)$	$(q_{cp_{r_2}}, a, L)$
q_{clean_2}	$(q_{clean_2}, a, R)(q_{clean_2}, a, L)$	$(q_{clean_2}, \#, R)$	(q_{move}, \square, L)	—
q_{move}	(q_{move_2}, \square, L)	—	—	—
q_{move_2}	(q_{move_2}, a, L)	—	(q_{copy}, a, R)	—

Erst wird a^1 erzeugt. Dann wird das vorherige Wort kopiert (mit $\#$ dazwischen), dann wird das Wort noch einmal kopiert, wobei jedoch $\#$ dazwischen gelöscht und das Wort 1 nach links bewegt wird. Dies geschieht wiederholt.