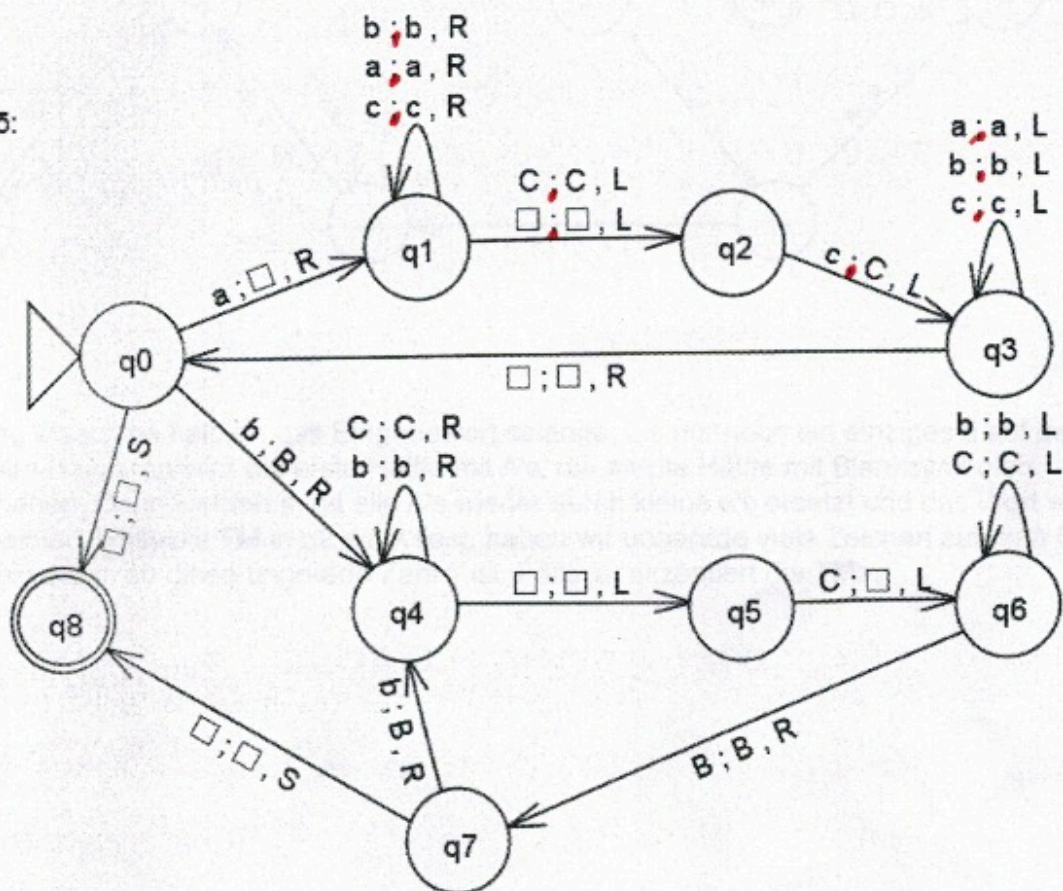


## Aufgabe 4

90%

1.

 $M = (\{q_0, \dots, q_8\}, \{a, b, c\}, \{a, b, c, B, C, \square\}, \delta, q_0, \square, \{q_8\})$ 
 $\delta$ :

Die Turing Maschine akzeptiert Wörter, die jeweils gleich viele a's, b's und c's enthalten. Sie überprüft zuerst, ob genauso viele a's wie c's im Wort sind, löscht dabei alle a's und überschreibt c's mit C's ( $q_0, \dots, q_3$ ). Dann wird die Anzahl von C's mit der Anzahl von b's verglichen ( $q_4, \dots, q_7$ ). Die C's werden gelöscht und die b's mit B's überschrieben. Wenn am Ende im Wort nur B's und keine C's stehen, haben wir genauso viele b's wie c's und die Turing Maschine geht in den Endzustand  $q_8$ .

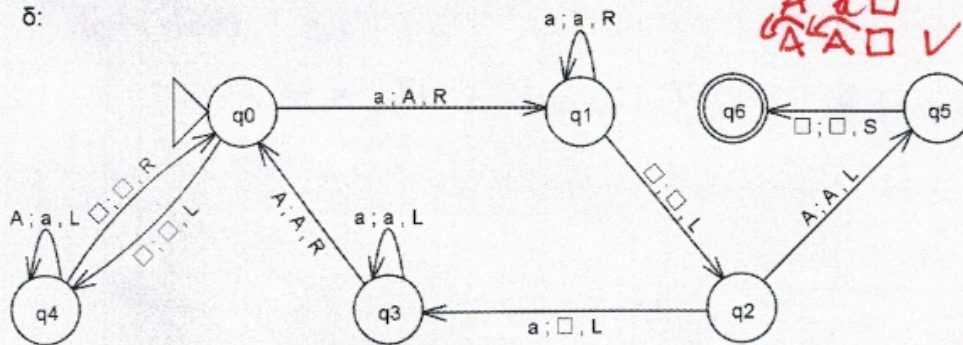
✓



2.

$M = (\{q_0, \dots, q_6\}, \{a\}, \{a, A, \square\}, \delta, q_0, \square, \{q_6\})$

$\delta$ :



Die Turing Maschine halbiert das Eingabewort solange, bis nur noch ein einziges a auf dem Band steht. Beim Halbieren wird die erste Hälfte mit A's, die zweite Hälfte mit Blanksymbolen überschrieben. Dann werden in q4 alle A's wieder durch kleine a's ersetzt und das Wort wird wieder halbiert. Falls die TM in q2 ein A liest, haben wir ungerade viele Zeichen auf dem Band. Wir überprüfen dann, ob diese ungerade Zahl 1 ist. Falls ja, akzeptiert die TM. ✓



## Aufg. 5

1.  $f$  ist berechenbar mit folgendem Algorithmus:

result  $\leftarrow 0$

while (now < 16.11.2042)

wait  $\leftarrow$  ~~next~~ :)

for  $i \leftarrow 1$  to 6 do

result  $\leftarrow$  result + lottozahlen[i]

// lottozahlen ruft  
die lottozahlen ab

if  $n = \text{result}$  do

return 1

else

return 0

$\rightarrow f$  ist durch einen Algorithmus  
in endlicher Zeit berechenbar

**Summe existiert bereits, einfach prüfen. (nun für uns unbekannt)**

2.  $g$  ist berechenbar, da es nach dem Satz d. Goldbach

unendlich viele Primzahlen gibt und es somit egal

wie groß  $n$  ist, immer auch eine  $n$ -te Primzahl geben

muß. Mit folgendem Algorithmus kann  $g$  berechnet werden:

$k \leftarrow 1$ ,  $l \leftarrow 1$

while  $k < n$

$l \leftarrow l + 1$

if  $l$  isPrime do

$k \leftarrow k + 1$

// ob eine Zahl eine Primzahl ist,  
ist berechenbar ~~ist~~

return  $l$



Aufg. 6

1.

20%  
Begründung / Beschreibung?

~~$x_3 := 0$~~

$x_4 := x_3 + 1;$  LOOP  $x_1$  DO  
LOOP  $x_2$  DO  $x_1 := x_1 - 1$  LOOP  $x_1 - x_2$  DO  
END  $x_3 := x_5 + 1;$  ex. nicht!  
LOOP  $x_1$  DO  $x_4 := x_5 + 0$  END

LOOP  $x_3$  DO  $P_1$  END;  
LOOP  $x_4$  DO  $P_2$  END

2.

LOOP  $x_2$  DO

$x_3 := x_1 + 0$

LOOP  $x_1$  DO

$x_4 := 0$

$x_5 := 1$  ← Warum?

LOOP  $x_1 - x_2$  DO

$x_4 := 1$

$x_5 := 0$  END

LOOP  $x_4$  DO

$x_3 := x_3 - x_2$

END

END

END

$x_1 > x_2$ : wird immer  
ausgeführt,  
also  $x_3 = 0!$

$x_0$  ist Rückgabewert!