

Mehr Turingmaschinen!

Aufgabe 1

Ein Palindrom ist ein Wort, dass man von vorne wie von hinten gleich lesen kann: ANNA, RENTNER, 011010110. Schreibe ein Turingmaschinen-Programm, dass Palindrome erkennt. Das Eingabealphabet sei $\{0, 1\}$. Falls ein Palindrom vorliegt, soll 1 ausgegeben werden, sonst 0.

Aufgabe 2

Schreibe ein Turingmaschinen-Programm, dass die Eingabebuchstaben in ihrer Reihenfolge invertiert: Also aus 1101110000 mach 0000111011, aus 001 mach 100 usw.

Wiederverwendung von Regeln

Erinnere dich an die Turing-Maschine, die Binärzahlen addiert:

Springe hinter erste Zahl:

```
jump1 0 0 R jump1
jump1 1 1 R jump1
jump1 $ $ R jump2
```

Springe hinter zweite Zahl und checke, ob sie größer Null ist:

```
jump2 0 0 R jump2
jump2 1 1 R jump2ok
jump2ok 0 0 R jump2ok
jump2ok 1 1 R jump2ok
jump2ok $ $ L dec
jump2 $ $ L toend1
```

Subtrahiere 1 von zweiter Zahl:

```
dec 1 0 L dec0
dec0 1 1 L dec0
dec0 0 0 L dec0
dec 0 1 L dec
dec0 $ $ L inc
```

Addiere 1 zu erster Zahl:

```
inc 0 1 L inc0
inc0 1 1 L inc0
inc0 0 0 L inc0
inc 1 0 L inc
inc $ 1 R jump1
inc0 $ $ R jump1
```

Springe vor die zweite Zahl, um später zu halten:
toend1 0 0 L toend1
toend1 1 1 L toend1
toend1 \$ \$ L toend2

Springe vor die erste Zahl, und halte:
toend2 0 0 L toend2
toend2 1 1 L toend2
toend2 \$ \$ R end

Aufgabe 3

Warum ähneln sich die Regeln für die Zustände toend1 und toend2 so sehr? Welche weiteren Ähnlichkeiten kann man im Programm feststellen? Beschreibe mit Worten, was diese ähnlichen Programmteile leisten.

Aufgabe 4

Braucht man diese vielen ähnlichen Zeilen? Fällt dir ein Trick ein, wie man Zeilen "wiederverwenden" kann? Behilf dir mit einem Übergangdiagramm.