

GTI Übungsblatt 8
Tutor: Marko Schmellenkamp
ID: MS1
Übung: Mi 16-18

Max Springenberg, 177792

8.1

8.1.1

(i)

$$f(1) = \perp$$

x_4 ist bei $x_1 = 1$ nach ablauf des ersten WHILE-Programms echt größer 0, damit endet das Folgende WHILE-Programm nie.

$$f(2) = 4$$

(ii)

$$f_P(n) = \begin{cases} 2^n, n \geq 2 \\ \perp, \text{sonst} \end{cases}$$

8.1.2

```
1:  x3 := 0;
2:  x4 := 1;
3:  x5 := x1;
4:  x6 := x2;

5:  IF x1 = 0 THEN GOTO 9;
6:  x1 := x1 ÷ 1;
7:  x3 := x3 + 1;
8:  IF x4 = 1 THEN GOTO 5;

9:  IF x3 = 0 THEN GOTO 15;

10: IF x3 = 0 THEN GOTO 20;
11: IF x2 = 0 THEN GOTO 18;
12: x2 := x2 ÷ 1;
13: x3 := x3 ÷ 1;
14: IF x4 = 1 THEN GOTO 10;

15: IF x2 = 0 THEN GOTO 20;
16: x2 := x2 ÷ 1;
17: IF x4 = 1 THEN GOTO 15;

18: x1 := x5;
19: IF x4 = 1 THEN GOTO 21;
20: x1 := x6;

21: HALT
```


8.2

8.2.1

Betrachte:

$$w \stackrel{\text{def}}{=} bab$$

Ersten 10 Konfigurationen:

$$\begin{aligned} & (q_b, (\epsilon, \triangleright, bab)) \vdash (q_b, (\triangleright, b, ab)) \\ & \vdash (q_r, (\triangleright \underline{b}, a, b)) \\ & \vdash (q_r, (\triangleright \underline{b}a, b, \epsilon)) \\ & \vdash (q_r, (\triangleright \underline{b}ab, \sqcup, \epsilon)) \\ & \vdash (q_l, (\triangleright \underline{b}a, b, \underline{b})) \\ & \vdash (q_l, (\triangleright \underline{b}, a, \underline{b}\underline{b})) \\ & \vdash (q_l, (\triangleright, \underline{b}, \underline{a}\underline{b}\underline{b})) \\ & \vdash (q_l, (\epsilon, \triangleright, \underline{b}\underline{a}\underline{b}\underline{b})) \\ & \vdash (q_b, (\triangleright, \underline{b}, \underline{a}\underline{b}\underline{b})) \end{aligned}$$

Erste Konfiguration in q_c :

$$(q_c, (\triangleright \underline{b}\underline{a}\underline{b}\underline{b}, \underline{b}, \sqcup))$$

8.2.2

Bedeutung der Zustände:

q_b

q_b liest den gesamten String von links nach rechts, bis er endet, bzw. ein \sqcup auftritt, oder ein b gelesen wird.

Wenn ein b gelesen wird, so wird dieses durch ein markiertes b (\underline{b}) ersetzt und in den Zustand q_l gewechselt.

Wenn ein \sqcup gelesen wird, hat der String keine unmarkierten b mehr und es wird in den Zustand q_c gewechselt.

q_c

q_c liest den gesamten String von links nach rechts und ersetzt alle markierten b mit unmarkierten, bis das Startsymbol erreicht wird.

In diesem Fall wird aufgehört den Lesekopf zu bewegen und in den Zustand h gewechselt.

q_r

q_r verschiebt den Lesekopf solange nach rechts, bis der String endet, bzw. ein \sqcup gelesen wird.

Dann wird dieses durch ein markiertes b ersetzt, bzw. ein markiertes b an den String hinten angehängen.

q_l

q_l verschiebt den Lesekopf solange nach links, bis das Startsymbol gelesen wird.
Dann wird in den Zustand q_b gewechselt.

h

Im Zustand h hält die Turingmaschine.

Aus der Bedeutung der Zustände geht hervor, dass:

Die Turingmaschine jedes b markiert und anschließend ein markiertes b an den String hinten anhängt, bis es keine unmarkierten b mehr gibt.

Abschließend werden alle markierten b durch normale b ersetzt und die TM hält.

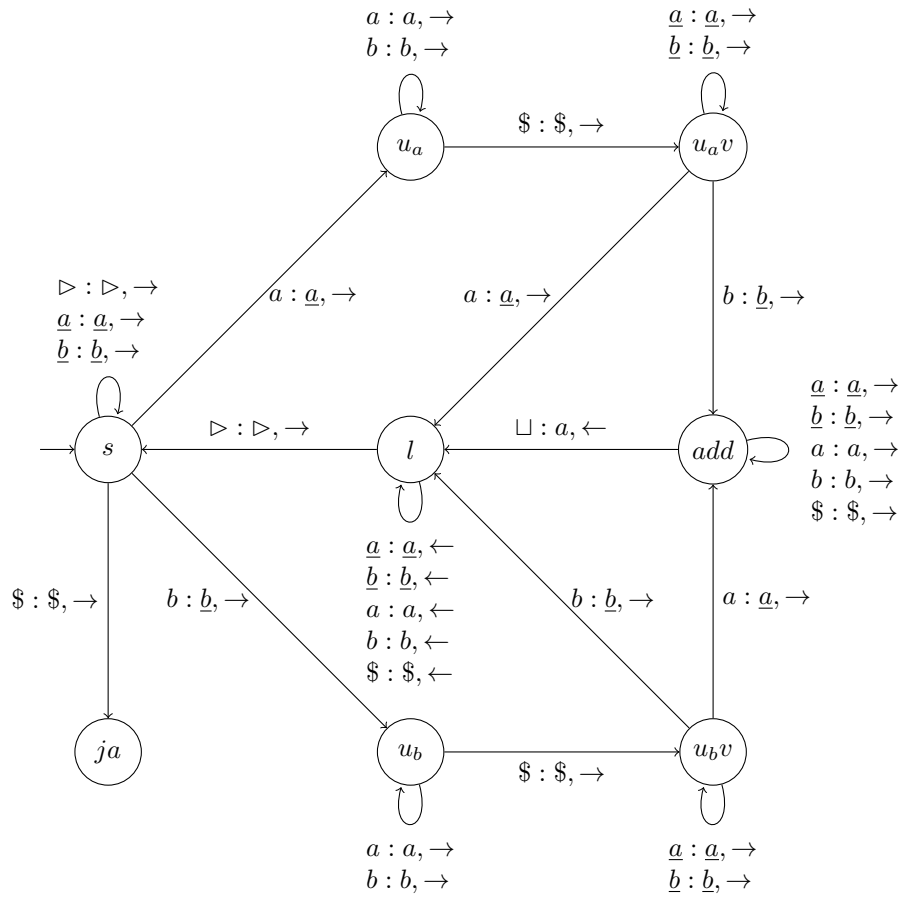
Da nach endlichen Operationen bei endlichen Eingaben alle b markiert worden sind hält die TM auch für alle Eingaben aus $\{a, b\}^*$ und weist damit keine Definitionslücken auf.

Für die Funktion f_M der TM bedeutet das, dass die Turingmaschine aus einem Wort $w \in \{a, b\}^*$ ein Wort $v \in \{a, b\}^*$, das aus w konkateniert mit $n = \#_b(w)$ b besteht, also der Form $v = wb^{n=\#_b(w)}$ macht.

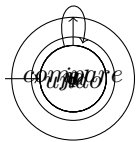
Dadurch ergibt sich dann auch die Funktion f_M zu:

$$f_M(w) = wb^{n=\#_b(w)}$$

8.3.1



Es wird davon ausgegangen, dass die Wörter unter korrekter Syntax eingegeben wurden. Alternativ kann vor der gegebenen TM auch eine TM durchlaufen werden, die nach der Syntax prüft, so eine wäre in etwa:

$$\begin{array}{l} \triangleright : \triangleright, \rightarrow \\ \underline{a} : \underline{a}, \rightarrow \\ \underline{b} : \underline{b}, \rightarrow \end{array}$$


s aus M_1 würde also unmittelbar auf ja aus M_2 folgen.

8.3.2

Es müsste jedes Zeichen gleich sein und ferner keine a angehängen werden.

Der Zustand add und alle Transitionen inzident zu diesem können entfernt werden.

Ferner führen nun alle Transitionen, die zuvor nach add führten nun nach $nein$ und statt markieren bleibt der Lesekopf stehen, da bereits ein ungleiches Zeichen ausreicht, damit das Wort nicht in der Sprache ist.

Die resultierende TM wäre damit:

