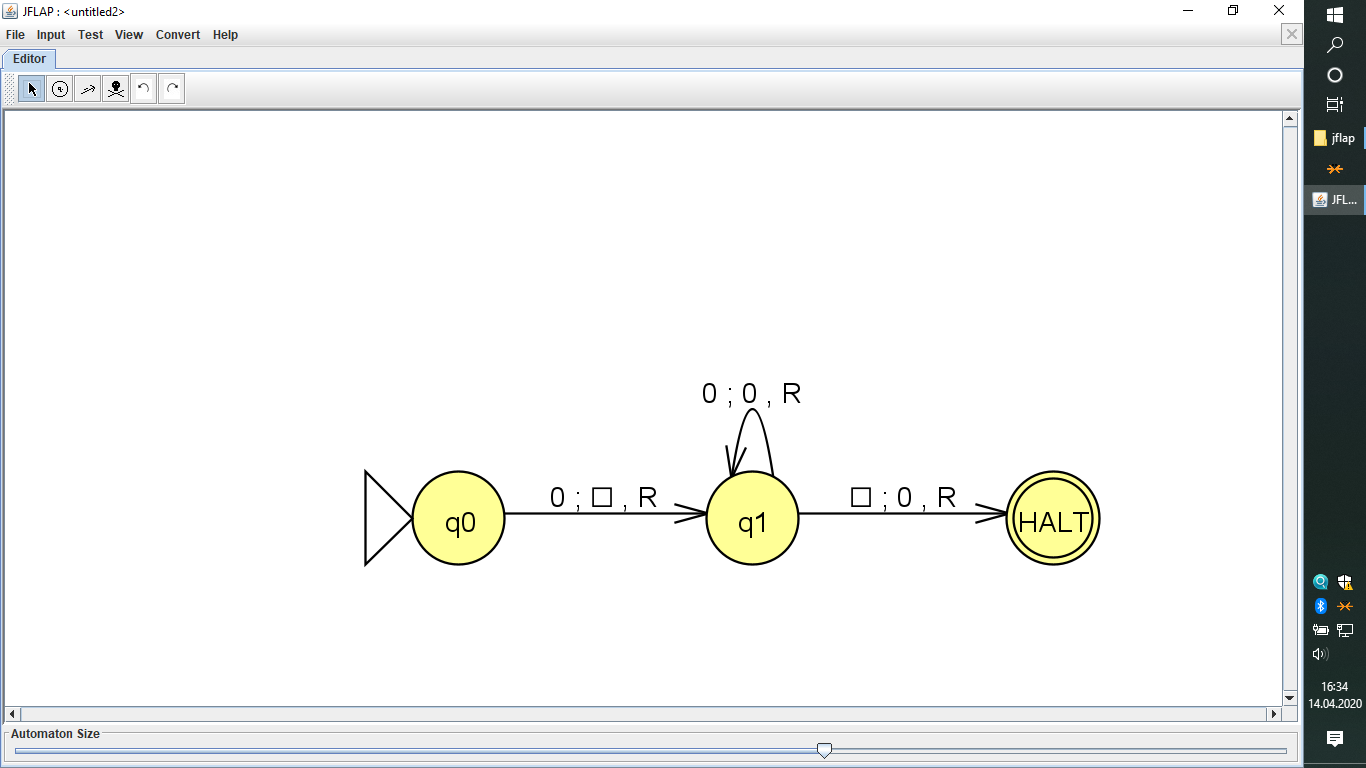
# Musterlösung Informatik 4 Stünder KW16 Di, 14 April

## Turing Maschinen, Teil 2

Es sollen schwierigere TMs eingeübt werden. Problem: TMs werden sehr schnell sehr kompliziert! Es gibt deshalb nur wenige Beispiele, die im Selbststudium zumutbar sind. Wir werden einfache Aufgabenstellungen verwenden!

## Aufgabe 1: Was machen diese TMs?

a) Teste mit den Inputs **„0“, „000“, „00000“**. Beschreibe in eigenen Worten.

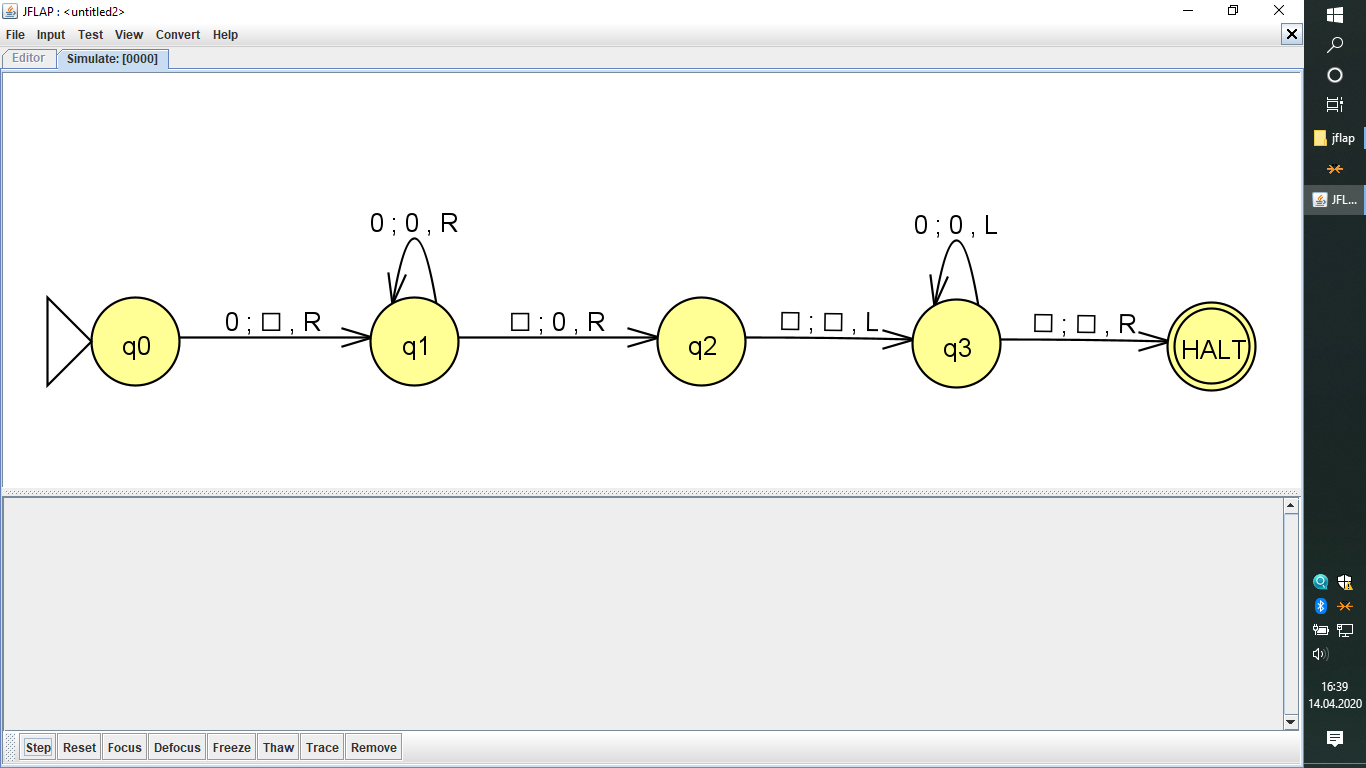
**Die TM verschiebt die „Nullen“ um eine weitere Stelle nach rechts. Allerdings ist das bei einem unendlich langen Tape ziemlich ‚egal‘. In Schritten aufgelöst:**

**q0 → q1: ‚0‘ einlesen, mit ‚◻‘ überschreiben, Tape-Head eins nach rechts**

**q1 → q1 Solange ‚0‘ vorhanden: ‚0‘ einlesen, mit ‚0‘ überschreiben, Tape-Head eins nach rechts**

**q1 → HALT: ‚◻ ‘ einlesen, mit ‚0‘ überschreiben, Tape-Head eins nach rechts**

b) Teste mit denselben Inputs wie in a) **(„0“, „000“, „00000“**). Beschreibe in eigenen Worten, was sich geändert hat.



**Die TM macht zunächst bis q2 genau das gleiche wie die obere TM, aber sie fährt mit dem Tape-Head an den Beginn der ‚0‘-Zeichenserie:**

**q0 → q1: ‚0‘ einlesen, mit ‚◻‘ überschreiben, Tape-Head eins nach rechts**

**q1 → q1 Solange ‚0‘ vorhanden: ‚0‘ einlesen, mit ‚0‘ überschreiben, Tape-Head eins nach rechts**

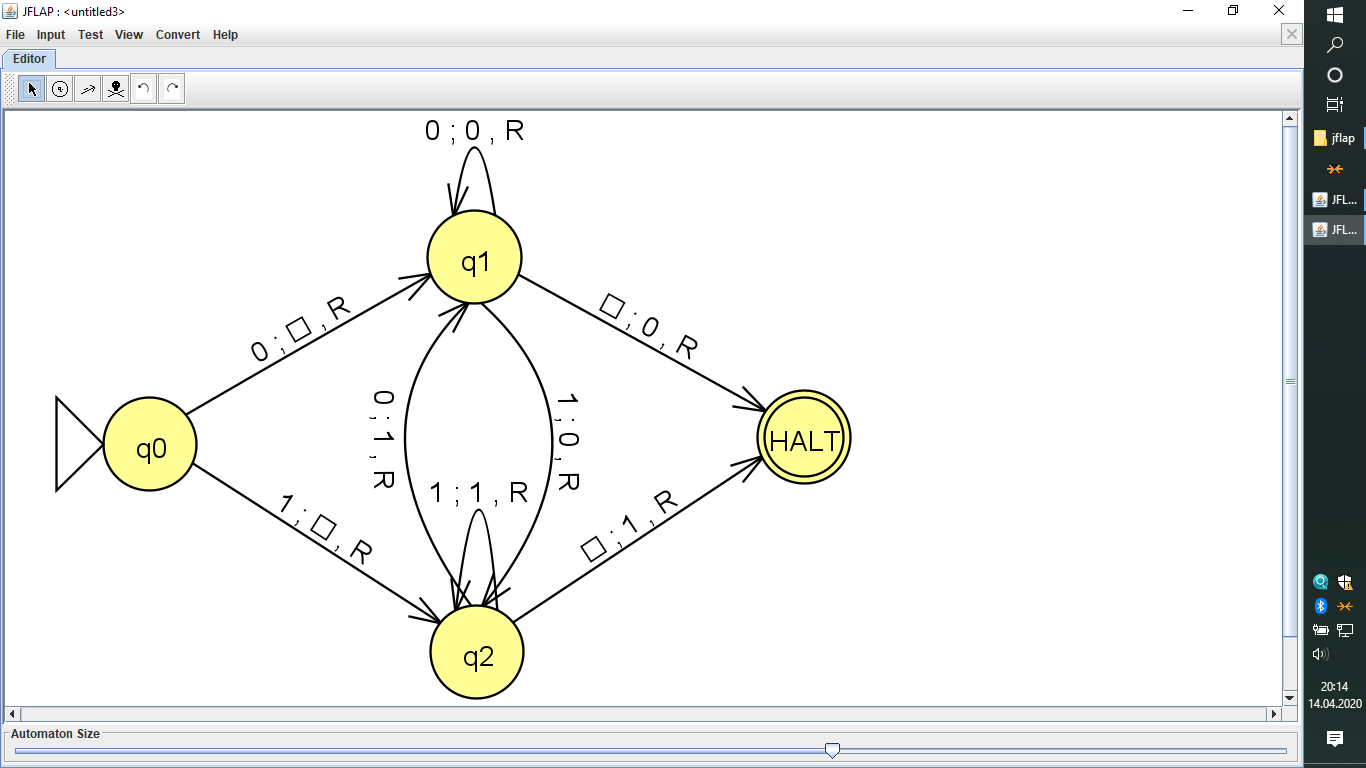
**q1 → q2: ‚◻ ‘ einlesen, mit ‚0‘ überschreiben, Tape-Head eins nach rechts**

**q2 → q3: ‚◻ ‘ einlesen, mit ‚◻‘ überschreiben, Tape-Head eins nach links**

**q3 → q3 Solange ‚0 ‘ vorhanden: ‚0 ‘ einlesen, mit ‚0‘ überschreiben, Tape-Head eins nach links**

**q3 → HALT: ‚◻ ‘ einlesen, mit ‚◻‘ überschreiben, Tape-Head eins nach rechts.**

c) Erweiterung der TM aus a): Teste mit den Inputs **„1001“, „0011“, „011“.**



**Die TM verschiebt den eingegebenen String um ein Zeichen nach rechts.**

## Aufgabe 2) Ein „alter Bekannter“

**Konstruiere eine TM für die Sprache L2 = {an bn | n ≥ 0}**

## Funktionsweise in Pseudocode:

1. Wenn das Symbol am Tape-Head gleich „a“, dann schreibe ◻***, ANSONSTEN VERWERFE!***
2. Bewege den Tape-Head solange nach rechts, bis der Tape-Head ein ◻ liest.
3. Bewege den Tape-Head eins nach links.
4. Wenn das Symbol am Tape-Head gleich „b“, dann schreibe ◻ ***, ANSONSTEN VERWERFE!***
5. Bewege den Tape-Head solange nach links, bis der Tape-Head ein ◻ liest.
6. Bewege den Tape-Head eins nach rechts.
7. Wenn das Symbol am Tape-Head gleich , ***DANN AKZEPTIERE!***

**1**

**2**

**3**

**4**

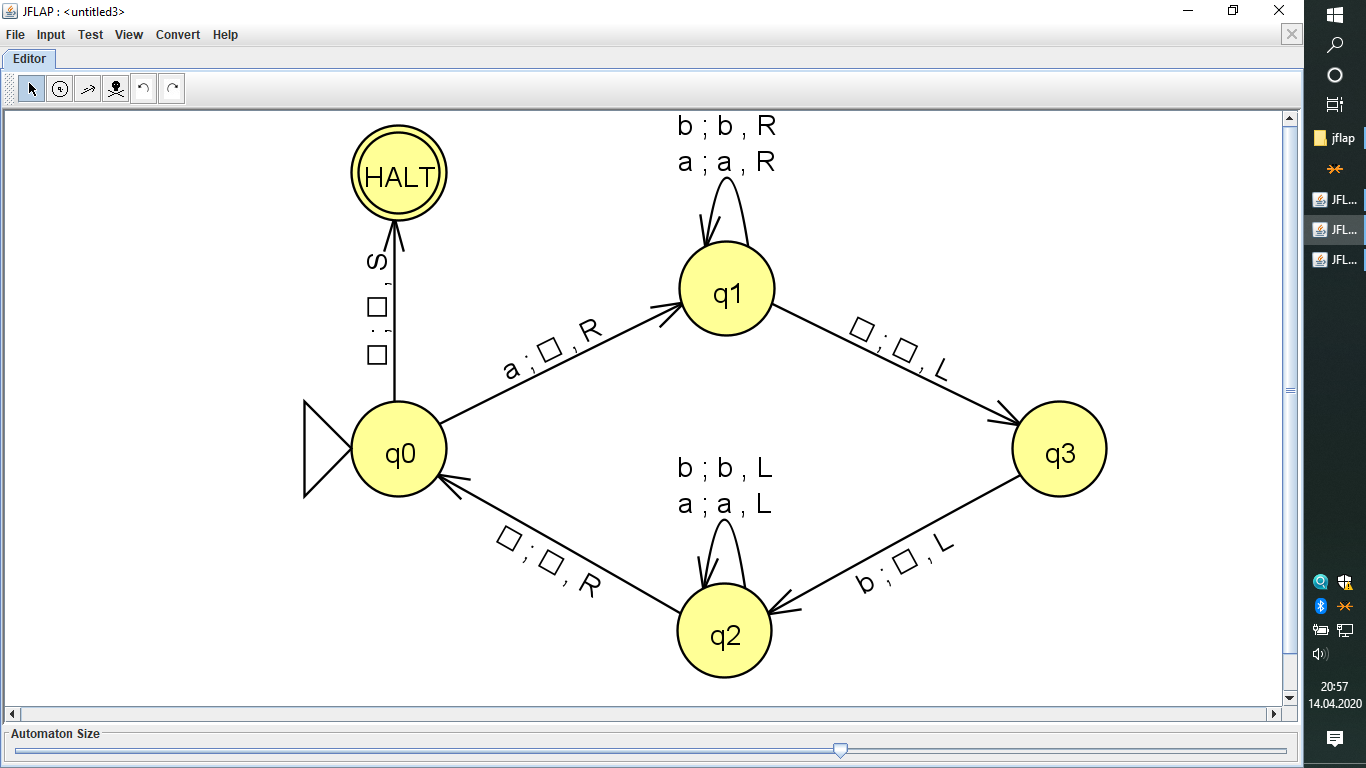
**5**

**6**

**8**

**7**

1. Beginne wieder bei 1 („GoTo 1“)



a) Teste mit den Inputs „**aabb**“, „**aaabb**“, „**aabbb**“. Beschreibe das Verhalten der TM mit eigenen Worten.

**Die TM arbeitet sich von den jeweiligen Endpunkten her in die Mitte vor: Zuerst wird das ‚a‘ zu Beginn weggenommen, danach geht der TM ans Ende des Wortes und nimmt dort ein ‚b‘ weg. Dann geht der Tape-Head wieder links zurück an den Beginn und der Zyklus „a weg – rechts ans Ende – b weg – links ans Ende“ verläuft von vorne. Akzeptiert wird ein Wort nur dann, wenn zuletzt kein Buchstabe mehr übrig ist.**

b) Ordne die einzelnen Punkte von 1. Bis 8. Den einzelnen Übergängen zu.

**Siehe Oben**

c) Für den Input „**aabb**“ fülle die folgende Einzelschritt-Tabelle aus:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ◻ | a | a | b | b | ◻ |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ◻ | ◻ | a | b | b | ◻ |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ◻ | ◻ | a | b | b | ◻ |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ◻ | ◻ | a | b | ◻ | ◻ |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ◻ | ◻ | a | b | ◻ | ◻ |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ◻ | ◻ | ◻ | b | ◻ | ◻ |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ◻ | ◻ | ◻ | b | ◻ | ◻ |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ◻ | ◻ | ◻ | ◻ | ◻ | ◻ |

## Aufgabe 3: Konstruktion einer einfachen TM

Konstruiere eine TM, welche eine beliebige Bitfolge invertiert, also z.B. aus „01101“ eine „10010“ erzeugt.

Ein Bild, das Uhr, Zeichnung enthält.

Automatisch generierte Beschreibung

**Abbildung 1: Lösung Sarah. Kurz, prägnant :-)**

## Aufgabe 4: Berechenbarkeit als zentraler informatischer Begriff, Bedeutung der TM

Es gibt im Internet sehr viele Erklärungen über den Begriff der „Turing-Berechenbarkeit“, allerdings sind diese meist auf Universitätsniveau und daher für uns leider völlig ungeeignet. Eine schöne Webseite findet sich allerdings hier:

<http://matheprisma.de/Module/Turing/index.htm?14>

*(wer den „vollen Einstieg“ möchte, kann auch hier beginnen:* [*http://matheprisma.de/Module/Turing/index.htm?1*](http://matheprisma.de/Module/Turing/index.htm?1)*)*

a) Erläutere in Stichworten:   
Was versteht man unter (Turing)-Berechenbarkeit, was ist das Halteproblem, und: *Was ist ein „fleißiger Biber*“?



**Berechenbarkeit: Berechenbar sind solche Probleme, welche von einer Turingmaschine gelöst werden können. Es gibt ebenfalls der Umkehrschluss: Ein Problem, welches nicht von einer TM gelöst werden kann, ist nicht lösbar.  
Wichtiges Kriterium für die Berechenbarkeit: Die Endlichkeit der Lösungsschritte; jedes lösbare Problem muss in einer bestimmten Anzahl an Schritten berechenbar sein, denn sonst findet die Berechnung womöglich nie ein Ende. Daraus resultiert das …**

**… Halteproblem: Es gibt kein Programm, welches die Frage beantworten kann, ob ein Computer rechnet, ob er abgestürzt ist oder ob er sich in einer Unendlichkeitsschleife befindet.**

**Fleissiger Biber:** <https://de.wikipedia.org/wiki/Fleißiger_Biber>