Turing Maschinen mögliche Prüfungsfragen

Turing Maschinen: Einführung

- Definieren Sie, was eine Turing-Maschine ist.

≙ Zustands-Maschine

Die Maschine, die 1936 von Alan Turing erfunden wurde, hat als Input ein ∞-langes Tape mit einem best. Alphabet und dem Zeichen '.' als Leerzeichen. Im Programm werden dann vers. Zustände von der Form:

" Zustand n
'read' 'write' 'direction' 'next state' ~~oder 'read' {c1, c2, ...} 'next state~~ "
(z.B. R, L, N, Pb (schreibe b))

definiert.

Mithilfe dieser eher schwierig zu programmierenden Maschinen konnten viele mathematischen Probleme gelöst werden, wie das Halting-Problem bzw. dass nicht alles berechenbar ist.

Dies ist eine Vereinfachung zur Programmierung für die Definition aber nciht nötig (aber auch nicht falsch).

bzw. erst einmal formuliert werden. Gerade der Begriff der Berechenbarkeit wurde so erst mathematisch fassbar.

- Erklären Sie die Begriffe «unäre», «binäre» und «dezimale» Darstellung von Zahlen.

unär: zum Beispiel: 1 1 1 ≙ 3 (→ Dezimalzahl)
eine Zahl als Folge von 1 dargestellt, dabei entspricht die Summe aller 1en der zugehörigen Dezimalzahl.

Beispiel: römische Zahlen

binär: zum Beispiel: 1 1 1 ≙ 7 (→ Dezimalzahl)
ein Zahl, deren Stellen den 2er Potenzen entsprechen (also 0. Stelle ≙ 2^0 bzw. 1. Stelle ≙ 2^1, usw.). Die Summe dieser 2er Potenzen entsprich der zugehörigen ~~Dezimalzahl~~.

Eine Zahl ist eine Zahl, unabhängig von Ihrer Darstellung.

dezimal: zum Beispiel 3
eine Zahl die als Kombination von Ziffern ∈[0, 9] angegeben wird.
Dabe entsprechen die Stellen vers. 10er-Potenzen also:
usw. | Hunderter (10^2) | Zehner (10^1) | Einer (10^0) | . | Zentel (10^{-1}) usw.

- Erklären Sie die Begriffe «Big-Endian» und «Little-Endian». Erwähnen Sie aus dem Alltag bekannte Beispiele.

Stelle — oder 10er-Potenz

Big-Endian: d.h. die erste ~~1~~ entspricht der höchsten 2er-Potenz: 100 ≙ 4
↳ also 'das grosse Ende' zuerst/am Anfang
→ die Bahnhofsuhr St. Gallen ist im Big-Endian Format

Und auch die gebräuchliche Notation im Zehnersystem.

Little Endian: d.h. die erste 1 entspricht der kleinsten 2er Potenz
also 2^0: 100 ≙ 1
↳ bzw. 'das kleine Ende' zuerst

Beispiel: Auf Deutsch werden Zahlen zwischen 13 und 99 "Little-Endian" ausgesprochen.

- Beschreiben Sie (d.h. liefern Sie eine Zustandsdefinition) eine Turing-Maschine, die

Input- und Output muss festgelegt werden. Format und Lesekopf-Position.

  1. unär 1 addiert.

② binär 1 addiert.
③ unär mit 2 multipliziert.
④ unär in binär oder umgekehrt umrechnet.

Annahme: little-endian-Format OK

#tape 111

④
```
start
. 1 stop
```
OK

②
```
start
1 0 R start
[.0] 1 R stop
```
OK

③
```
start
. x L read

read L
1 x R start
. . N write

write
x 1 R write
. . N stop
```
Input-Format noch festlegen.

Sollte 'R' anstatt 'N' sein. Sonst liest write sofort den '.'

Maschine hält am Ende der Zahl

unär → binär Little-Endian

④
```
start
1 x R z1

z1
. ! R add

add
1 0 R add
[.0] 1 L left

left L
. . R del

del
1 x R right
! ! L del2

right
! ! R add

del2 L
x . L del2
. . N stop
```

Universelle Turing Maschine

- Definieren Sie, was eine «universelle Turing-Maschine (UTM)» ist.

Eine TM die als Input eine andere TM und deren Input hat.
Die UTM simuliert dementsprechend die beg. TM.

- Wenn man mit Binärzahlen mit einer beschränkten Anzahl Stellen arbeitet, warum kann 111111111 als einer Darstellung von -1 aufgefasst werden?

Betrachtet man 3 Stellen. Die folgende TM würde dann 1 zu jeder Zahl addieren
↳ little-Endian

```
start
[.0] 1 R stop
1 0 R start
```

Aber bei 111 würde ...000 1... herauskommen, bei Betrachtung von 3 Stellen also 000 ≙ 0. Wenn aber "111 + 1 = 000" dann muss die Zahl 111 der Dezimalzahl -1 entsprechen.

OK, weitere Betrachtungsweise mit der Anordnung der 8 (bzw. 2^n) Zahlen auf einem Kreis. 7 vorwärts heisst dann das gleiche wie 1 rückwärts.

- Beschreiben Sie, wie ein Zustand einer TM für eine UTM codiert werden könnte und geben Sie ein Beispiel an.

Der Zustand wird mit '>.' initialisiert. Danach folgt ein read-symbol, ein write-symbol, die Richtung und der Folgezustand in binär codierung dabei hat der erste Zustand die Nummer 0:

Plus ein Punkt um die Binärzahl abzuschliessen.

Zustand 0
1 1 R Zustand 0
. 1 L Zustand 1

} wäre übersetzt: >. 11 R 0 .. 1 L 1

Halting Problem

- Beschreiben Sie, was das «Halting Problem» ist und welche praktische Konsequenzen das hat.

Das Halting-Problem untersucht ob es eine TM gibt, die den output hat ob eine weitere TM hält oder eine ∞ Schleife hat. Dabei kommt heraus, dass es so eine TM nicht geben kann.

mit gegebenem Input

Das Halting-Problem ist der Beweis, dass es keine Maschine gibt die untersuchen kann, ob eine andre funktioniert und damit ist auch nicht alles berechenbar.

- Beweisen Sie das «Halting Problem».

Annahme:
Sei H eine TM mit Input (T,I)
Output von H:
- T hält auf I
- T hält nicht auf I
- H hält nicht (ausschliessbar)

Was macht H mit...

TM 'S'
S soll als Input (T,I) haben
Output von S: Mit Hilfe von H wird untersucht, ob T auf I hält.
- wenn T auf I hält dann hält S nicht (Endlosschleife)
- wenn T auf I nicht hält, dann hält S

FOLGE: ~~Sei H eine TM~~ mit Input (S,(S,I)):
1. Fall: H meldet, dass S hält
↳ a.n. S auf (S,I) hält nicht. ↯

2. Fall: H meldet, dass S nicht hält
↳ a.n. S auf (S,I) hält ↯

Demensprechend muss die Annahme falsch sein.

Busy Beaver

- Sei S(n,m) die maximale Anzahl Schritte, die eine TM mit n Zuständen und einem Alphabet von m Zeichen, auf einem leeren Band machen kann und am Schluss noch stoppt. Zeigen Sie, dass S(n,m) eine nicht-berechenbare Funktion ist.

Die Funktion S(n,m) ist berechenbar. (Existieren tut die Funktion, weil sie ja soeben eindeutig definiert wurde).

Annahme ~~es existierte eine Funktion~~ S(n,m).

Dann wäre das Halteproblem lösbar.

Sei nämlich m=2. Dann müsste man nur $\sum(3n+6)$ berechnen und wenn die TM nach $\sum(3n+6)$ Schritten nicht anhält, hält sie nie.
Das Problem ist, dass $\sum(3n+6)$ schon für n=6 nicht berechenbar ist und für n=7 nicht mal schätzbar.
Zudem ist das Halteproblem nicht lösbar wie oben bewiesen wurde, dementsprechend führt diese Annahme auf einen Wiederspruch und ist somit falsch. ~~Es existiert keine Funktion~~ S(n,m). ist nicht berechenbar.

Folgende TM würde das Halteproblem lösen (zumindest für Maschinen, die auf leerem Band starten):

Man simuliert die zu untersuchende Maschine für maximal S(n,m) Schritte. Wenn diese bis dann nicht angehalten hat, hält sie nie mehr.