Lecture1_BeKo

Introduction

The "Hello World" - Problem

- Goal: Develop a Program E with the following specs:
  - Input: Program P
  - Output: Print: "Top", when P contains the string "Hello, World", else print: "Flop"
- Remark: E has "A type of higher order",(which means that the input is text from a program P)
- Ex:

```
main() {
  printf("Hello, world");
}
```

- Does there exist a program E for a special input P? -> Yes
- Does there exist a program E for every program P? -> Not possible

Finite Automata

- A deterministic finite automata (DFA) is a quintuple M = {Z,$\Sigma, \delta, z\_0, E\}$ with:

  - Z is a non empty, finate group of states
  - Σ is a non empty, finite alphabet of input characters with $Z \cap E = 0$
  - $\delta : Z \times \Sigma$ -> Is a partial transfer function
  - For M we define a partial function $\delta : Z \times \Sigma^*$ -> Z is inductive for all $z \epsilon Z$:
    - $\delta(z, \epsilon) := z$
    - $\forall x \epsilon \Sigma^* \quad \delta(z, ax) := \hat{\delta}(\delta(z, a), x)$ in case $\delta(z, a)! = \perp$
  - A DFA M = {Z,$\Sigma, \delta, z\_0, E\}$ accepts a word $W \epsilon E$ if $\hat{\delta}(z_0, w) \epsilon E$
  - The words from M are an accepted language:
    - T(M):= $\{x \epsilon \Sigma^* \mid \hat{\delta}(z_0, x) \epsilon E\}$

Example:

- M = $(\{z_0, z_1, z_2\}, \{0, 1\}, \delta, z_0, \{z_2\})$

δ	z_0	z_1	z_2
0	z_0	z_1	z_2
1	z_0	z_1	z_2

- Boundaries of Finite Automata
  1. $\{w\epsilon\{0,1\}^*|w$ is a binary representation of a even number$\}$ -> Is Finite due to there the language being limited by the amount of 0 and 1
  2. { $a^n\ b^n \mid 0 <= n <= 1000\}$ -> I True because there is a boundary from 0 to 1000
  3. { $a^n\ b^n \mid n >= 0\}$ -> Is False finite because the automata can not count the amount of a's and b's because there is no boundary to m
  4. $\{(abc)^n \mid n >= 0\}$ -> Is True because the automata can count all be counted due to all nodes being counted at the same time # 1.1
  5. $\{a^n b^m c^k \mid n, m, k >= 1\}$ -> Is True because #TODO: Ask tutor 1.2
  6. $\{a^n b^n c^n \mid n >= 0\}$ -> False for same reason as 2.

Turing Machines

- A Turing Machine (DTM) is a septuple group M = $(Z, \Sigma, \Gamma, \delta,\ z_0, \sqcup, E)$ with:
  - Z is a non empty, finate group of states
  - Σ is a non empty, finite alphabet of input characters with $Z \cap E = 0$
  - $\Gamma \supseteq \Sigma$ is a Work and/or ribbon alphabet with $\Gamma \cap Z = 0$
  - $\delta : (Z \backslash E) \times \Gamma -> Z \times \{L, R, N\}$ is the partial transfer function
    - This is the most important aspect of the Turing machine
    - $\delta : (Z \backslash E)$ Is the NON-end state of the Turing machine
    - Γ: This is the current read state of the Turing machine (what is currently read)
    - Z defines the start of a new state
    - Γ is the new symbol that will be written in the ribbon (We are overwriting the currently written symbol)
    - $\{L, R, N\}$ is the possibility of movement after the machine is finished writing
  - $z_0 \epsilon Z$ is the start state
  - $\sqcup \epsilon \Gamma \setminus \Sigma$ is the blank symbol (its the current symbol)
  - E $\subseteq$ Z is a group of end states

Interpreting the actions of a Turing machine

- When M reads a while in state z we get $\delta(z, a) = (z', a', p)$ so that:
  - M goes to state z'
  - We overwrite a with a'
  - The Head of moves to p (Left, Right or Neutral)

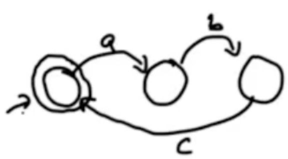
Look at 1.3 for example

Turing Machine Configuration

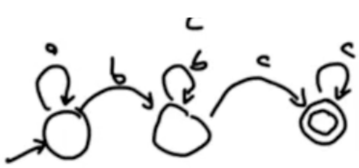
- M = $(Z, \Sigma, \Gamma, \delta,\ z_0, \square, E)$ is a TM. A configuration M is a word azb with a,b$\epsilon\Gamma$ and z ϵZ
- A start configuration of a word $x\epsilon\Sigma^*$ is z_0x
- Let k = $a_1....a_m\mathbf{z}b_1......b_n$ be a configuration (if n = 0, then $b_1 := \square$) Then the following is true:
  - $k \vdash_M^0 k$
  - $k \vdash_M^1 a_1.....a_m\mathbf{z}'cb_2....b_n$ if $\delta(z, b_1) = (z', c, N)$
  - $k \vdash_M^1 a_1.....a_m\mathbf{z}'b_2....b_n$ if $\delta(z, b_1) = (z', c, R)$
  - $k \vdash_M^1 a_1.....a_m - 1\mathbf{z}'a_mcb_2....b_n$ if $\delta(z, b_1) = (z', c, L)$ and m>0
  - $k \vdash_M^1 \mathbf{z}'\square cb_2....b_n$ if $\delta(z, b_1) = (z', c, L)$ and m=0
- k is holding (meaning it doesn't have a vaild configuration) if $\delta(z, b_1) = \perp$
- k is accepted if $z\epsilon E$
- Furthermore let $k \vdash_M^i +1k'$ <=> $\exists_q k \vdash_M^1 q \vdash_M^i k'$ for all i and k $k \vdash_M^* k'$ <=>$\exists_{i\epsilon N} k \vdash_M^i k'$

Turing Machine example

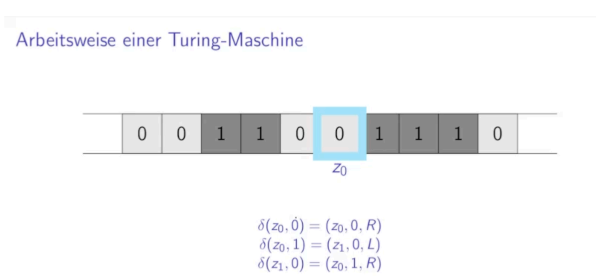
- Look 1.5



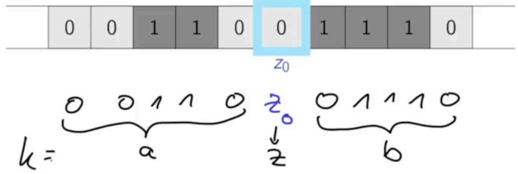
1.1:



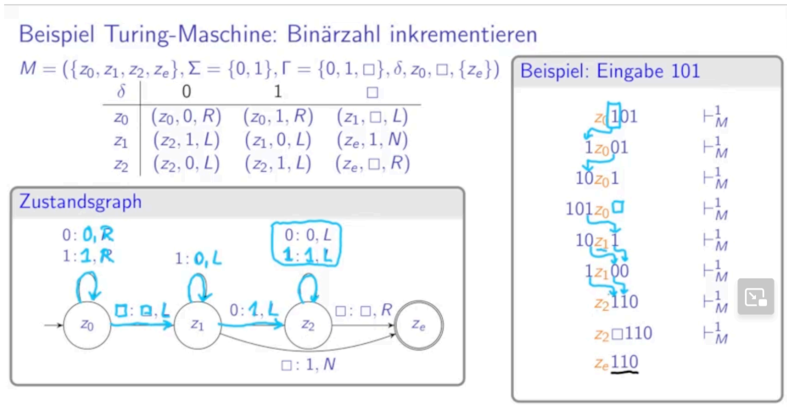
1.2:



1.3


0 0 1 1 0 0 1 1 1 0
z0
0 0 1 1 0 z0 0 1 1 1 0
k = a z b

1.4


Beispiel Turing-Maschine: Binärzahl inkrementieren
M = ({z0, z1, z2, ze}, Σ = {0, 1}, Γ = {0, 1, □}, δ, z0, □, {ze})
δ | 0 | 1 | □
z0 | (z0, 0, R) | (z0, 1, R) | (z1, □, L)
z1 | (z2, 1, L) | (z1, 0, L) | (ze, 1, N)
z2 | (z2, 0, L) | (z2, 1, L) | (ze, □, R)
Zustandsgraph
0: 0, R
1: 1, R
1: 0, L
0: 0, L
1: 1, L
□: □, L
0: 1, L
□: □, R
□: 1, N
z0
z1
z2
ze
Beispiel: Eingabe 101
z0101 ⊢M
1z001 ⊢M
10z01 ⊢M
101z0□ ⊢M
10z11 ⊢M
1z100 ⊢M
z2110 ⊢M
z2□110 ⊢M
ze110

1.5