Edin Gusa
23435

# CS407 HW1

1)

| Label | Condition | TM | |
|---|---|---|---|
| >A | — | R.B | → Start scanning |
| B | $\sigma = \#$ | L.#.h | → When # is reached, halt. |
| | $\sigma = \$$ | R.C | → If we see a $, left shift the rest of the word |
| | else | R.B | → Else, keep scanning |
| C | $\sigma = x \neq \#$ | L.x.R.C | → Left shift operation |
| | $\sigma = \#$ | L.#.$L_{\#}$.A | |

4.1.4) The turing machine M semidecides the language $L = \{a^n.a \; ; \; n \text{ is even}\}$

4.1.7) $M = (Q, \Sigma, \delta, s, H)$

$Q = \{A, B, C, h\}$

$\Sigma = \{a, b, \#, \Diamond\}$

$s = A$

$H = \{h\}$

Initial Configuration: $(A, \Diamond \underline{\#} w)$

$\delta$:

| Label | Condition | TM |
|---|---|---|
| >A | — | R.B |
| B | $\sigma = a$ | R.C |
| | $\sigma \neq a$ | R.B |
| C | $\sigma = a$ | h |
| | $\sigma \neq a$ | R.B |

4.1.12) $(\Diamond \# aabb \underline{\#}) \vdash (\Diamond \# aabb \underline{b}) \vdash^2 (\Diamond \# aab \# \underline{\#}) \vdash^2 (\Diamond \# aab \underline{\#} b) \vdash (\Diamond \# aab \underline{b} b)$

$\vdash (\Diamond \# aa \underline{b} bb) \vdash^3 (\Diamond \# aa \# bb \underline{b}) \vdash^2 (\Diamond \# aa \underline{b} bbb) \vdash (\Diamond \# a \underline{a} bbbb) \vdash^5 (\Diamond \# a \underline{a} bbbba)$

$\vdash (\Diamond \# \underline{a} abbbba) \vdash^5 (\Diamond \# \underline{a} abbbbaa) \vdash (\Diamond \underline{\#} aabbbbaa) \vdash (\Diamond \# aabbbbaa \underline{\#})$

4.2.2)

a) $>h_{NO}$

b) $>R \xrightarrow{\#} h_{YES}$, $R \xrightarrow{\{a,b\}} h_{NO}$

c) $>R \xrightarrow{a} R \xrightarrow{\#} h_{YES}$, $R \xrightarrow{\{b,\#\}} h_{NO}$, $R \xrightarrow{\{a,b\}} R$

d) $>R \xrightarrow{b} h_{NO}$, $R \xrightarrow{\#} h_{YES}$, $R \xrightarrow{a} R$

4.1.12 - New) $>L \xrightarrow{\sigma = x \neq \#} R.x.L$, $L \xrightarrow{\sigma = \#} R.\#.R_{\#}.h$ ⇒ Right-shift Turing Machine

$(\Diamond \# aabb \underline{\#}) \vdash (\Diamond \# aabb \underline{b}) \vdash^2 (\Diamond \# aabb \underline{b}) \vdash^2 (\Diamond \# aa \underline{b} bb)$

$\vdash^2 (\Diamond \# aabb \underline{b} b) \vdash^2 (\Diamond \# aa \underline{b} bb) \vdash^2 (\Diamond \# aa \underline{b} bb) \vdash^2 (\Diamond \# \underline{a} abb) \vdash^2 (\Diamond \# aabb)$

$\vdash^2 (\Diamond \# \underline{a} abb) \vdash^2 (\Diamond \# \underline{a} abb) \vdash^2 (\Diamond \# \# aabb) \vdash (\Diamond \# \# aabb \underline{\#})$

4.3.1)

a) A two-way infinite tape TM M with $H = \{h\}$ is said to semidecide a language $L \subseteq \Sigma_0^*$ if:

$(s, \underline{\#} w \#) \vdash_M^* (h, u \underline{a} v)$, iff $w \in L$

b) A k-tape TM M is said to compute a function $f: \Sigma_0^* \to \Sigma_0^*$ if:

$(s, \Diamond \underline{\#} w, \Diamond \underline{\#}, \ldots, \Diamond \underline{\#}) \vdash_M^* (h, u_1 \underline{a_1} v_1, \ldots, u_k \underline{a_k} v_k)$ iff $u_i = \Diamond$ ; $a_i = \#$, $v_i = f(w)$

Edin Gusa
23435

4.3.2) a) $M = (Q, \Sigma, \delta, s, H)$

$Q$ = finite set of states

$\Sigma$ = finite symbol alphabet $(\Sigma = \Sigma_0 \cup \# \cup \Diamond)$

$s$ = initial state $(s \in Q)$

$H$ = set of halt states $(H \subseteq Q)$

$\delta : Q - H \times \Sigma^k \to Q \times \{\rightarrow, \leftarrow, \Sigma\}^k$

constraints:

1 - $\delta(q, \Diamond, \ldots, x) = (q', \rightarrow, \ldots, \{\rightarrow, \leftarrow, \Sigma\})$

2 - cannot write $\Diamond$

b) Instantaneous Description: $(q, u \underset{1}{a_1} v_1, \ldots \underset{k}{a_k} v_k)$ if all the heads are at different positions

$(q, u \underset{1}{a_1} v_1 \ldots \underset{s,t,\beta}{a_m} v_m \ldots \underset{k}{a_k} v_k)$ if multiple heads are at the same position

Start Convention: $(s, \Diamond \underset{1,\ldots,k}{\#} w)$ where $w \in \Sigma_0^*$

4.3.6) $M = (Q, \Sigma, \delta, s, H)$

$Q$ = finite set of states

$\Sigma$ = finite symbol alphabet $(\Sigma = \Sigma_0 \cup \# \cup \Diamond \cup \Diamond_h \cup \Diamond_v)$

$s$ = initial state $(s \in Q)$

$H$ = set of halt states $(H \subseteq Q)$

$\delta : Q - H \times \Sigma \to Q \times \{\rightarrow, \leftarrow, \uparrow, \downarrow, \Sigma\}$

constraints:

1 - $\delta(q, \Diamond_h) = (q', \uparrow)$

2 - $\delta(q, \Diamond_v) = (q', \rightarrow)$

3 - cannot write $\Diamond$, $\Diamond_h$, $\Diamond_v$

Instantaneous Description: $(q, \begin{bmatrix} x & w & z \\ u & \underline{a} & v \\ s & y & t \end{bmatrix})$ where

$a$ = the symbol under the head
$u \in \Sigma^*$ = the string to the left of the head
$v \in \Sigma^*$ = the string to the right of the head
$w \in \Sigma^*$ = the string above the head
$y \in \Sigma^*$ = the string below the head
$x \in \Sigma^* \times \Sigma^*$ = the 2D string top left of the head
$z \in \Sigma^* \times \Sigma^*$ = the 2D string top right of the head
$s \in \Sigma^* \times \Sigma^*$ = the 2D string bottom left of the head
$t \in \Sigma^* \times \Sigma^*$ = the 2D string bottom right of the head

Start Convention: $(s, \begin{bmatrix} \Diamond_v & \# & w \\ \Diamond_v & \underline{\#} & \# \\ \Diamond & \Diamond_h & \Diamond_h \end{bmatrix})$

where $w$ is the 2D input word. $w \in \Sigma_0^* \times \Sigma_0^*$

Computational Notation:

$(q, \begin{bmatrix} x & w & z \\ u & \underline{a} & v \\ s & y & t \end{bmatrix}) \vdash_M (q', \begin{bmatrix} x' & w' & z' \\ u' & \underline{b} & v' \\ s' & y' & t' \end{bmatrix})$, a single step computation

$(q, \begin{bmatrix} x & w & z \\ u & \underline{a} & v \\ s & y & t \end{bmatrix}) \vdash_M^n (q', \begin{bmatrix} x' & w' & z' \\ u' & \underline{b} & v' \\ s' & y' & t' \end{bmatrix})$, an n step computation

$(q, \begin{bmatrix} x & w & z \\ u & \underline{a} & v \\ s & y & t \end{bmatrix}) \vdash_M^* (q', \begin{bmatrix} x' & w' & z' \\ u' & \underline{b} & v' \\ s' & y' & t' \end{bmatrix})$, a finite step computation

Edin Gus.
23435

## Deciding a language:

A TM $M$ with a 2-dimensional tape with $H=\{h_{YES}, h_{NO}\}$ is said to decide $L \subseteq \Sigma_0^* \times \Sigma_2^*$ if:

$$(s, \begin{bmatrix} \Diamond_v & \# & w \\ \Diamond_v & \underline{\#} & \# \\ \Diamond & \Diamond_h & \Diamond_h \end{bmatrix}) \vdash_M^* (h_{YES}, \begin{bmatrix} x & w & z \\ u & \underline{a} & v \\ s & y & t \end{bmatrix}) \quad \text{if } w \in L$$

$$(s, \begin{bmatrix} \Diamond_v & \# & w \\ \Diamond_v & \underline{\#} & \# \\ \Diamond & \Diamond_h & \Diamond_h \end{bmatrix}) \vdash_M^* (h_{NO}, \begin{bmatrix} x & w & z \\ u & \underline{a} & v \\ s & y & t \end{bmatrix}) \quad \text{if } w \notin L$$

## Simulating by a standard TM:

A TM $M$ with a 2-dimensional tape and with contents:


can be simulated by a TM $M'$ with a 1-dimensional tape and with contents:

$\$ \Diamond_h 1 \Diamond_v 2 3 \Diamond_h 4 5 6 \Diamond_v 7 8 9 \Diamond_h 10\ 11\ 12\ 13\ 14\ 15\ \Diamond_v - - -$

This is achieved by mapping every 2-dimensional:

$\Sigma$: $\Sigma$ (remains the same)

$\uparrow$: Move right, until you hit $\Diamond_h$ or $\Diamond_v$ and count the steps you took (say $k$ steps)
- if you hit $\Diamond_h$: move $k+1$ more steps
- if you hit $\Diamond_v$: move $k$ more steps

$\downarrow$: Move left until you hit $\Diamond_h$ or $\Diamond_v$ and count the steps you took (say $k$ steps)
- if you hit $\Diamond_h$: move $k-1$ more steps
- if you hit $\Diamond_v$: move $k$ more steps

$\rightarrow$: Move right until you hit $\Diamond_h$ or $\Diamond_v$ and count the steps you took (say $k$ steps)
- if you hit $\Diamond_h$: move $k$ more steps
- if you hit $\Diamond_v$: move $k+1$ more steps

$\leftarrow$: Move left until you hit $\Diamond_h$ or $\Diamond_v$ and count the steps you took (say $k$ steps)
- if you hit $\Diamond_h$: move $k$ more steps
- if you hit $\Diamond_v$: move $k-1$ more steps

* The counting can be done on a second tape. As we have seen in class a 2-tape TM can be simulated by a single-tape TM.

* 1 step of computation for an input length $n$ will take $O(n)$ time because for each step you take at most $2k+1$ steps and $k<1$. Therefore $t$ steps for input length $n$ will have $O(nt)$ time.