Lab10-Turing Machine

Algorithm and Complexity (CS214), Xiaofeng Gao, Spring 2020.

∗ If there is any problem, please contact TA Yiming Liu.
∗ Name: Futao Wei   Student ID: 518021910750   Email: weifutao2019@gmail.com

1. Design a one-tape TM M that computes the function $f(x,y) = x \mod y$, where x and y are positive integers ($x > y$). The alphabet is $\{1, 0, \Box, \rhd, \lhd\}$, and the inputs are x 1's, $\Box$ and y 1's. Below is the initial configuration for input $x = 7$ and $y = 3$. The result $z = f(x,y)$ should also be represented in the form of z 1's on the tape with the pattern of $\rhd 111 \cdots 111 \lhd$.

Initial Configuration

	$\rhd$	1	1	1	1	1	1	1	$\Box$	1	1	1	$\lhd$	
	$\uparrow$ q_S													

(a) Please describe your design and then write the specifications of M in the form like $\langle q_S, \rhd \rangle \to \langle q_1, \rhd, R \rangle$. Explain the transition functions in detail.

(b) Please draw the state transition diagram.

(c) Show briefly and clearly the whole process from initial to final configurations for input $x = 7$ and $y = 3$. You may start like this:

$(q_s, \underline{\rhd}1111111\Box111\lhd) \vdash (q_1, \rhd\underline{1}111111\Box111\lhd) \vdash^* (q_1, \rhd1111111\underline{\Box}111\lhd) \vdash (q_2, \rhd1111111\Box\underline{1}11\lhd)$

(Note that for simplicity, we write $(q_1, \rhd\underline{1}111111\Box111\lhd) \vdash^* (q_1, \rhd1111111\underline{\Box}111\lhd)$ if the corresponding transaction repeats on multiple inputs with the same state.)

Solution.

(a) To compute the function $f(x,y)$, we're going to decrement x and y by 1 repeatedly. When y turns 0, we'll recover it to the initial value y_0. When x finally turns 0 with the current value of y being y', we can obtain $f(x,y) = y_0 - y'$. The following is the specifications of M.

Start: trivial

$$\langle q_S, \rhd \rangle \to \langle q_1, \rhd, R \rangle$$

Step 1: $x \leftarrow x - 1$ until $x = 0$

$$\langle q_1, 1 \rangle \to \langle q_2, \rhd, R \rangle$$
$$\langle q_1, \Box \rangle \to \langle q_6, \rhd, R \rangle$$

Step 2: move to y part on the tape

$$\langle q_2, 1 \rangle \to \langle q_2, 1, R \rangle$$
$$\langle q_2, \Box \rangle \to \langle q_3, \Box, R \rangle$$

Step 3: skip the 0's; $y \leftarrow y - 1$ until $y = 0$

$$\langle q_3, 0 \rangle \to \langle q_3, 0, R \rangle$$
$$\langle q_3, 1 \rangle \to \langle q_4, 0, L \rangle$$
$$\langle q_3, \lhd \rangle \to \langle q_5, \lhd, L \rangle$$

Step 4: move back to x part on the tape

$$\langle q_4, 0\rangle \to \langle q_4, 0, L\rangle$$
$$\langle q_4, \Box\rangle \to \langle q_4, \Box, L\rangle$$
$$\langle q_4, 1\rangle \to \langle q_4, 1, L\rangle$$
$$\langle q_4, \triangleright\rangle \to \langle q_1, \triangleright, R\rangle$$

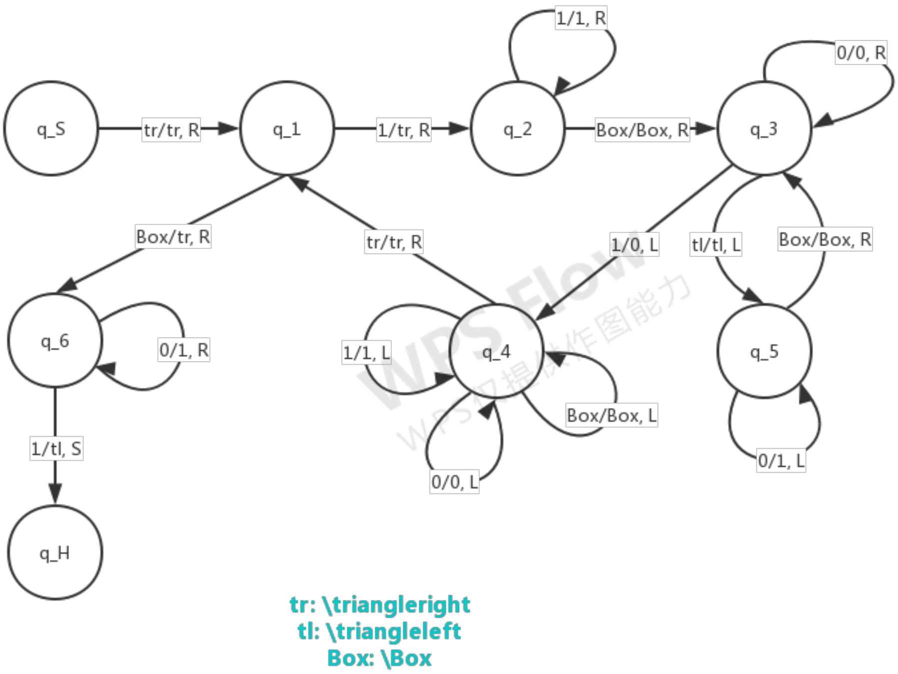
Step 5: recover y_0 1's when $y = 0$

$$\langle q_5, 0\rangle \to \langle q_5, 1, L\rangle$$
$$\langle q_5, \Box\rangle \to \langle q_3, \Box, R\rangle$$

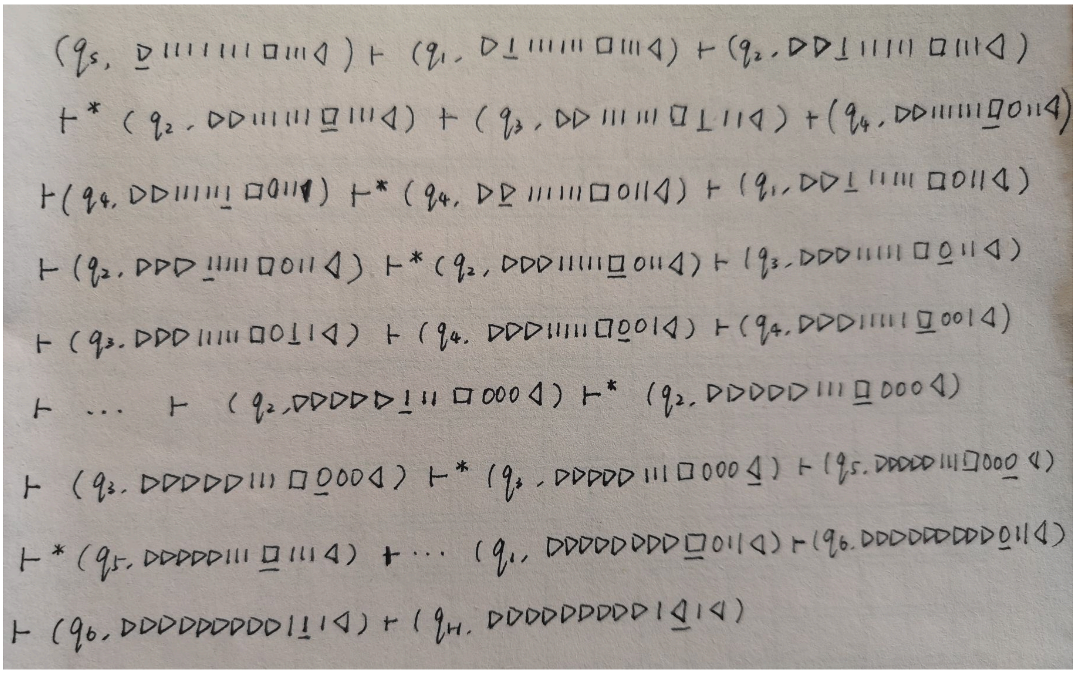
End: write down the result

$$\langle q_6, 0\rangle \to \langle q_6, 1, R\rangle$$
$$\langle q_6, 1\rangle \to \langle q_H, \triangleleft, S\rangle$$

(b)



(c)



$\square$

2. Assume there's a Turing Machine M using alphabet $\Gamma : \{\triangleright, \square, a, b, \cdots, z\}$. We can simulate M by a Turing Machine $\tilde{M}$ using alphabet $\tilde{\Gamma} : \{\triangleright, \square, 0, 1\}$. Please transform the instruction $\langle q, i\rangle \to \langle q', j, R\rangle$ in M into its corresponding form in $\tilde{M}$.

Solution. We can represent the 26 letters with a sequence of 0's and 1's whose length is 5. For generality, denote i as $a_{i1}a_{i2}a_{i3}a_{i4}a_{i5}$ and j as $a_{j1}a_{j2}a_{j3}a_{j4}a_{j5}$, where $a_k \in \{0, 1\}$. The following is the corresponding instructions.
Step 1: read the sequence denoting i

$$\begin{gathered}\langle q, a_{i1}\rangle \to \langle q_1, a_{i1}, R\rangle\\ \langle q_1, a_{i2}\rangle \to \langle q_2, a_{i2}, R\rangle\\ \langle q_2, a_{i3}\rangle \to \langle q_3, a_{i3}, R\rangle\\ \langle q_3, a_{i4}\rangle \to \langle q_4, a_{i4}, R\rangle\end{gathered}$$

Step 2: return to the start

$$\begin{gathered}\langle q_4, a_{i5}\rangle \to \langle q_5, a_{i5}, L\rangle\\ \langle q_5, a_{i4}\rangle \to \langle q_6, a_{i4}, L\rangle\\ \langle q_6, a_{i3}\rangle \to \langle q_7, a_{i3}, L\rangle\\ \langle q_7, a_{i2}\rangle \to \langle q_8, a_{i2}, L\rangle\end{gathered}$$

Step 3: write the sequence denoting j

$$\begin{gathered}\langle q_8, a_{i1}\rangle \to \langle q_9, a_{j1}, R\rangle\\ \langle q_9, a_{i2}\rangle \to \langle q_{10}, a_{j2}, R\rangle\\ \langle q_{10}, a_{i3}\rangle \to \langle q_{11}, a_{j3}, R\rangle\\ \langle q_{11}, a_{i4}\rangle \to \langle q_{12}, a_{j4}, R\rangle\\ \langle q_{12}, a_{i5}\rangle \to \langle q', a_{j5}, R\rangle\end{gathered}$$

$\square$

3. **Wireless Data Broadcast System.** In a Wireless Data Broadcast System (WDBS), data items are repeatedly broadcasted in cycle on different channels. Denote $D = \{d_1, d_2, \cdots, d_k\}$ as data items, each d_i with length l_i (as time units), and $\mathbf{C} = \{C_1, C_2, \cdots, C_n\}$ as broadcasting channels. Fig. 1 illustrates a WDBS with 25 data items and 4 channels. Once a channel finishes broadcasting current cycle, it will repeat these data again as a new cycle. E.g., a possible broadcasting sequence of C_1 could be $\{d_6, d_{12}, d_1, d_{18}, d_7, d_6, d_{12}, d_1, d_{18}, d_7, \cdots\}$

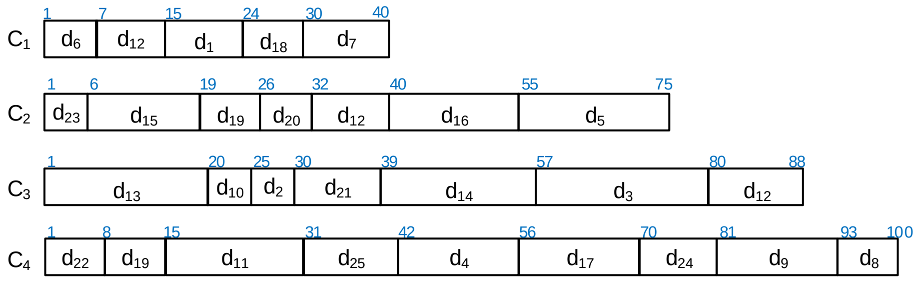


图 1: An Example Scenario of Wireless Data Broadcast System.

If a mobile client requires a subset of data items $D_q \subseteq D$ from this WDBS, he/she must access onto one channel, wait for the appearance of one required item, and switch to another channel if necessary. Each "switch" requires one time slot. For example, Lucien wants to download $\{d_1, d_3, d_5\}$, as shown in Fig. 2. He firstly accesses onto C_1 at time slot 1, then download d_1, d_3 respectively during time slots 2 to 5, and then switch to C_3 at time slot 6 (note that he cannot download d_5 from C_2 because of the switch constraint), and download d_5 during time slots 7 to 8. We define *access latency* as the period when a client starts downloading, till the time he/she finishes. As a result, the overall access latency for Lucien is 7 in this example.

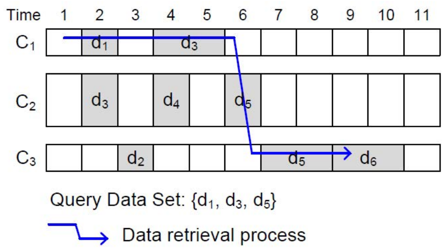


图 2: An Example Scenario of Query of a Client.

Each operation (download/wait/switch) needs energy consumption. To conserve energy, a client hopes to use minimum amount of energy to download all required items in D_q, which means that he/she waits to minimize both access latency and switch numbers. Unfortunately, these two objectives conflict with each other naturally. Fig. 3 exhibits such a scenario. To download $D_q = \{d_1, d_2, d_3, d_4\}$, if we start from C_2, in Option 1 we can switch to C_1 for d_1 immediately after downloading d_3, return back to C_2 for d_4, and to C_1 again for d_2. Such option costs 3 switches and 7 access latency. While in Option 2, we stay at C_2 lazily for d_3 and d_4, and then switch to C_1 for d_2 and d_1. Such option costs 1 switches and 12 access latency.

Once we want to minimize two conflictive objectives simultaneously, we have three possible ways (similar as Segmented Least Squares told in Dynamic Programming Lecture). Now it is your turn to complete the formulation of this optimization, we name it as Minimum Constraint Data Retrieval Problem (MCDR), with the following sub-questions.

(a) If we add an additional switch parameter h, please define the MCDR (Version 1) completely as a search problem.

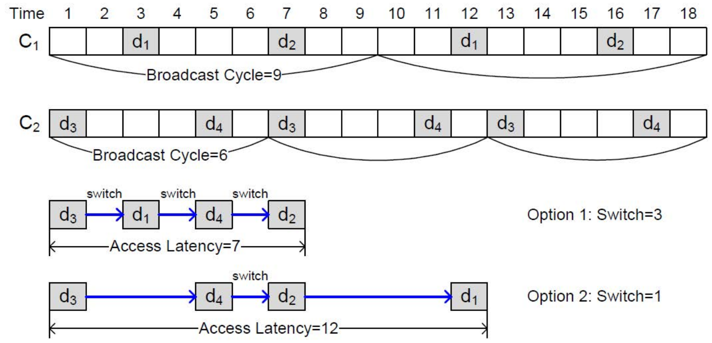


图 3: Confliction between Access Latency and Switch Number.

(b) If we add an additional latency parameter t, please define the MCDR (Version 2) completely as a search problem.

(c) If we set dimensional parameters α to switch number, and β to access latency, we can combine two objectives together linearly as a new concept "cost". Please define the Minimum Cost Data Retrieval Problem (MCDR, Version 3) correspondingly.

(d) Please give the decision versions of sub-questions (a), (b) and (c).

Solution.
Input:

- Data item set $D = \{d_1, d_2, \cdots, d_k\}$, each d_i with length l_i as time units
- Broadcasting channel set $\mathbf{C} = \{C_1, C_2, \cdots, C_n\}$, each C_i with data sequence S_i to be repeatedly broadcasted
- Broadcasting starting time t_i for each C_i, which is defined to obtain the time differences between two channels
- The data item set $D_q \subseteq D$, which is needed by the client

(a) Output (search version):
  - The minimum switch cost $h_{\min}$
  - A legal sequence of (C_k, d_k) denoting the order of downloading d_k from C_k, which achieves the minimum switch cost $h_{\min}$

  Output (decision version):

$$ans = \begin{cases} 1, & \text{if } P \text{ is true} \\ 0, & \text{if } P \text{ is false} \end{cases}$$

  where P: There exists a legal sequence of (C_k, d_k) which achieves a switch cost less than h_0 (some specific value).

(b) Output (search version):
  - The minimum latency $t_{\min}$
  - A legal sequence of (C_k, d_k) denoting the order of downloading d_k from C_k, which achieves the minimum latency $t_{\min}$

Output (decision version):

$$ans = \begin{cases} 1, & \text{if } P \text{ is true} \\ 0, & \text{if } P \text{ is false} \end{cases}$$

where P: There exists a legal sequence of (C_k, d_k) which achieves a latency less than t_0 (some specific value).

(c) Output (search version):

- The minimum combinational cost $(\alpha h + \beta t)_{\min}$
- A legal sequence of (C_k, d_k) denoting the order of downloading d_k from C_k, which achieves the minimum combinational cost $(\alpha h + \beta t)_{\min}$

Output (decision version):

$$ans = \begin{cases} 1, & \text{if } P \text{ is true} \\ 0, & \text{if } P \text{ is false} \end{cases}$$

where P: There exists a legal sequence of (C_k, d_k) which achieves a combinational cost less than $\alpha h_0 + \beta t_0$ (some specific value).

$\square$