CS 373: Theory of Computation

Sariel Har-Peled and Madhusudan Parthasarathy

Discussion : High-level TM design

7 April 2008

1 Questions on homework?

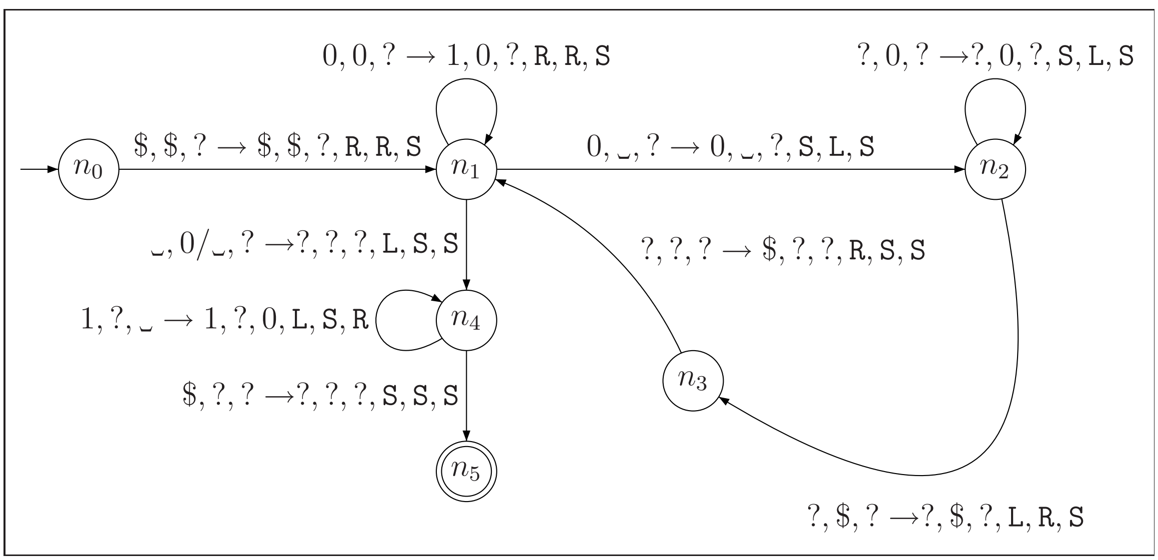
Any questions? Complaints, etc?

2 High-level TM design

2.1 Modulo

Question 2.1 *Design a TM that given* $\$0^{a}$ *on tape* $\circledast_1$ *and* $\$0^{b}$ *on tape* $\circledast_2$*, writes* $0^{a \bmod b}$ *on tape* $\circledast_3$*.*

Solution: The idea is to check off every b characters from $\circledast_1$ and copy the remaining characters to $\circledast_2$. We assume that ␣ is the blank symbol on tapes and S indicates that the head remains stationary.



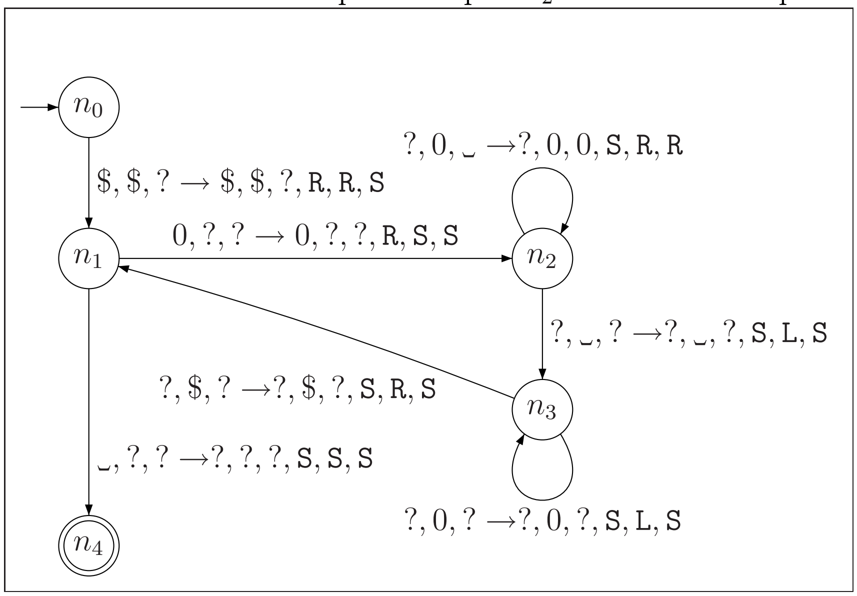
The basic idea is that immediately after we move the head of $\circledast_2$ back to the beginning of the tape (state n_2), we write a \$ on the first tape (i.e., transition from n_3 to n_1). Thus, conceptually, every time this loop is being performed a block of b charactere of 0 are being chopped of $\circledast_1$.

To use this box as a template in our future designs (less and more like Macros in C++), we name this **Mod**($\circledast_1$, $\circledast_2$, $\circledast_3$). ■

2.2 Multiplication

Question 2.2 *Design a* TM *that given* $\$0^a$ *on tape* ✇$_1$ *and* $\$0^b$ *on tape* ✇$_2$*, writes* 0^{ab} *on tape* ✇$_3$*.*

Solution: The idea is to attach a copies of tape ✇$_2$ at the end of tape ✇$_3$.



To use this box as a template in our future designs (less and more like Macros in C++), we name this **Mult**(✇$_1$, ✇$_2$, ✇$_3$). ■

2.3 Binary Addition

Question 2.3 *Design a* TM *that given* w_1^R *on tape* ✇$_1$ *and* w_2^R *on tape* ✇$_2$*, writes* w_3^R *on tape* ✇$_3$*, where* $w_1, w_2, w_3 \in \{0,1\}^*$ *and* w_3 *is the binary addition of* w_1 *and* w_2*.*

Thus, if ✇$_1 = \mathtt{01}$ *(i.e., this is the number* $\mathtt{10}_2 = 2$ *and* ✇$_2 = \mathtt{1011}$ *(i.e., this is the binary number* $\mathtt{1101}_2 = 13$ *then the output should be* $\mathtt{1111}$ *(which is* 15*).*

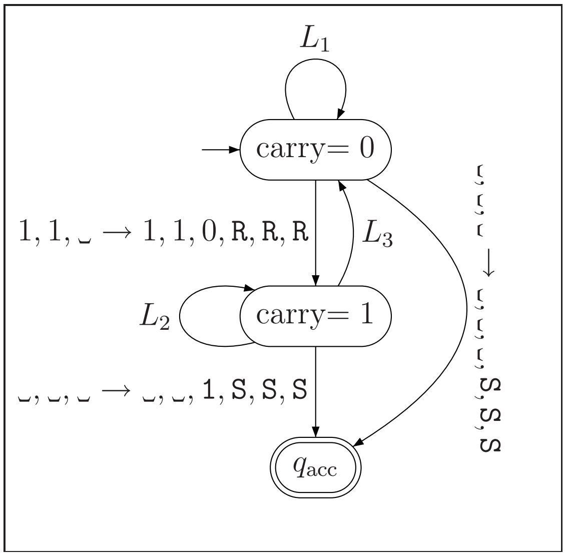
Solution: We sum starting from the least significant digit, the normal procedure. See Figure **??**. Here, a transition of the form $0, 1, ␣ \rightarrow 0, 1, 0, \mathtt{R}, \mathtt{R}, \mathtt{R}$ stands for the situation where the TM reads 0 on the first tape, 1 on the second tape and ␣ on the third tape, next it writes 0, 1 and 0 to these three tapes respectively, and move the three heads to the right. ■

2.4 Quadratic Remainder

Question 2.4 *Design a* TM *that given* $0^a\$0^b$ *leaves only* 0^c *on the tape where* $c \in \mathbb{N}_0$ *is the smallest number such that* $c^2 \equiv a \bmod b$*. If such a* c *does not exist, the TM must reject.*

Solution: To solve this problem we assume that we have some other macros in addition to those we have built already.

- **CLEAR**(t): write ␣ on all the non-␣ characters of the tape t till it encounters a ␣. Then returns the head to the beginning of t.



L_1 :
$0, 0, ␣ \to 0, 0, 0, \mathtt{R}, \mathtt{R}, \mathtt{R}$
$1, 0, ␣ \to 1, 0, 1, \mathtt{R}, \mathtt{R}, \mathtt{R}$
$0, 1, ␣ \to 0, 1, 1, \mathtt{R}, \mathtt{R}, \mathtt{R}$
$␣, 0, ␣ \to ␣, 0, 0, \mathtt{S}, \mathtt{R}, \mathtt{R}$
$␣, 1, ␣ \to ␣, 1, 1, \mathtt{S}, \mathtt{R}, \mathtt{R}$
$0, ␣, ␣ \to 0, ␣, 0, \mathtt{R}, \mathtt{S}, \mathtt{R}$
$1, ␣, ␣ \to 1, ␣, 1, \mathtt{R}, \mathtt{S}, \mathtt{R}$

L_2 :
$1, 0, ␣ \to 1, 0, 0, \mathtt{R}, \mathtt{R}, \mathtt{R}$
$0, 1, ␣ \to 0, 1, 0, \mathtt{R}, \mathtt{R}, \mathtt{R}$
$1, 1, ␣ \to 1, 1, 1, \mathtt{R}, \mathtt{R}, \mathtt{R}$
$␣, 1, ␣ \to ␣, 1, 0, \mathtt{S}, \mathtt{R}, \mathtt{R}$
$1, ␣, ␣ \to 1, ␣, 0, \mathtt{R}, \mathtt{S}, \mathtt{R}$

L_3 :
$0, 0, ␣ \to 0, 0, 1, \mathtt{R}, \mathtt{R}, \mathtt{R}$
$␣, 0, ␣ \to ␣, 0, 1, \mathtt{S}, \mathtt{R}, \mathtt{R}$
$0, ␣, ␣ \to 0, ␣, 1, \mathtt{R}, \mathtt{S}, \mathtt{R}$

Figure 1: The TM for the binary addition.

- **EQ**(✇$_1$, ✇$_2$): checks whether the number of non-space characters on ✇$_1$ and ✇$_2$ are the same.
- **GREQ**(✇$_1$, ✇$_2$): checks whether the length of string of tape ✇$_1$ is more than or equal to that of ✇$_2$.
- **COPY**(✇$_1$, ✇$_2$): copies content of tape ✇$_1$ to tape ✇$_2$.

Note that our TM just needs to check for c among $\{0, 1, 2 \cdots, b-1\}$. We should serialize our macros in the following way (instead of a diagram, we write pseudo-code which is more readable). We use 7 tapes. Figure **??** depicts the resulting TM. ■

3 MTM

Question 3.1 *Show that an* **MTM** *(an imaginary more powerful TM) whose head can read the character under the head and the character to the left of the head (if such a character does not exist, it will read a ␣, i.e. blank character) and can just rewrite the character under the head, is equivalent to a normal TM.*

Solution: First of all it is obvious that an **MTM** can simulate a TM since it can ignore the extra information that it can read using its special head.

Now observe that a TM can simulate an **MTM** this way: For making a move using the transition function of **MTM**, the TM that simulates it must read the character under the head (which a normal TM can) and the character to the left of head (which a normal TM can't). What the simulating TM does is that it remembers the current state of **MTM** in its states (note that we have done several times, this kind of "remembering finite amount of information inside states by redefining states, e.g. extending them to tuples" in class), brings the head to the left and reads that character and remembers it inside its states, moves

```
copy from beginning till $ from ✇_1 to ✇_2
copy from $ on tape ✇_1 till end to ✇_3

/*** ✇_2 = 0^a and ✇_2 = 0^b ***/

CLEAR(✇_1)
CLEAR(✇_7)
Mod(✇_2, ✇_3, ✇_5)
/*** ✇_5 = 0^(a mod b)        ***/

do
    if EQ(t_7, t_5) then accept.
    if GREQ(✇_1, ✇_3) then reject.
    add one 0 at the end of ✇_1
    COPY(✇_1, t_4)
    Mult(✇_1, t_4, t_6)
    Mod(t_6, ✇_3, t_7)
while true
```

Figure 2: The algorithm for $c^2 = a \bmod b$.

the head to the right, so now head is in its original place and our TM knows the missed character and can perform the correct move using **MTM**'s transition function. ■