VIII

TM's to Compute Number-Theoretic Functions

- Recall ... a Number Theoretic Function:

$f : \mathbb{N} \to \mathbb{N}$ or $f : \mathbb{N}^n \to \mathbb{N}$

- unary notation – a number will be represented as a string of 1's

1 = 1
11 = 2
111 = 3
⋮

HOWEVER zero is a legitimate value

therefore the number n will be represented by a string of $n+1$ 1's., i.e.

1 = 0
11 = 1
111 = 2
1111 = 3
⋮

the additional 1 will be referred to as a representational 1.

(first or last 1 ... does not matter).

Computing Number Theoretic Functions

- Representing a pair of numbers on a Tm tape:

... B 11 B 1111 B ...

(2,4)

EXAMPLE 1.5.1 Suppose that the Turing machine of Figure 1.5.1 is started scanning the leftmost of three *1*s, say, representing natural number 2. Plainly it will write a single additional *1* to the left and halt scanning four *1*s representing natural number 3. On the other hand, if started scanning a representation of 3, it will halt scanning a representation of 4. More generally, if started scanning an unbroken string of $n+1$ *1*s **representing natural number** n, for any $n \geq 0$, it will halt scanning an unbroken string of $n+2$ *1*s **representing natural number** $n+1$. This leads us to say that this Turing machine *computes* the number-theoretic function defined by

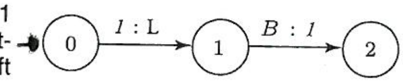


Figure 1.5.1 A Turing machine that computes the successor function.

$$f(n) = n + 1$$

Function f is just the unary successor function.

B 111 B ⊢ B 111 B ⊢ B 1111 B

q_0 q_1 q_2

initial tape final tape

$f(2) = 3$

B 11111 B ⊢ B 11111 B ⊢ B 111111 B

q_0 q_1 q_2

initial tape final tape.

$f(4) = 5$

- More generally ... $f(n) = n+1$ // unary successor function

Number Theoretic Functions

- How should Tm M behave when f is a partial function?

EXAMPLE 1.5.2 Consider the partial number-theoretic function defined by

$$f: \mathcal{N} \to \mathcal{N}$$

$$f(n) = \sqrt{n}$$

- Naturally, as a number-theoretic function, f is only defined when n is a perfect square (and is represented by n+1 1's).

- example:

When n is 9 we would have

B1111111111B // ten 1's representing n=9 as initial tape

$\uparrow$
q_0

then final tape would look as:

B1111B // four 1's representing n=3 as final tape $\sqrt{9} = 3$

$\uparrow$
q_f

Partial Number Theoretic Functions

- Continuing with $f: \mathbb{N} \to \mathbb{N}$
  $f(n) = \sqrt{n}$

- What should our Tm M do when n is not a perfect square?

for example when $n = 5$

B 111111 B // six 1's
↑
q_0
initial tape

neither

B 111 B // n=2
↑
q_f

nor

B 1111 B // n=3
↑
q_f

is acceptable.

- M should not halt in a value-representing configuration.

Any of

B 11**1B or B 1x1zB or BB
↑ ↑ ↑
q_f q_f q_f

would do.

Or M could not halt at all in these situations.

Number-Theoretic Functions

more formally:

DEFINITION 1.5: *Deterministic Turing machine M computes (unary) partial number-theoretic function f* provided that:

(i) If M is started scanning the leftmost *1* of an unbroken string of $n + 1$ *1s* on an otherwise blank tape, where function f is defined for argument n, then M halts scanning the leftmost *1* of an unbroken string of $f(n) + 1$ *1s* on an otherwise blank tape.

(ii) If M is started scanning the leftmost *1* of an unbroken string of $n + 1$ *1s* on an otherwise blank tape, where function f is undefined for argument n, then M does not halt in a value-representing configuration; that is, either M does not halt at all or, if M does halt, it does not halt scanning the leftmost *1* of an unbroken string of *1s* on an otherwise blank tape.

Example 1.5.3 A Turing machine that computes $f(n) = 2n$

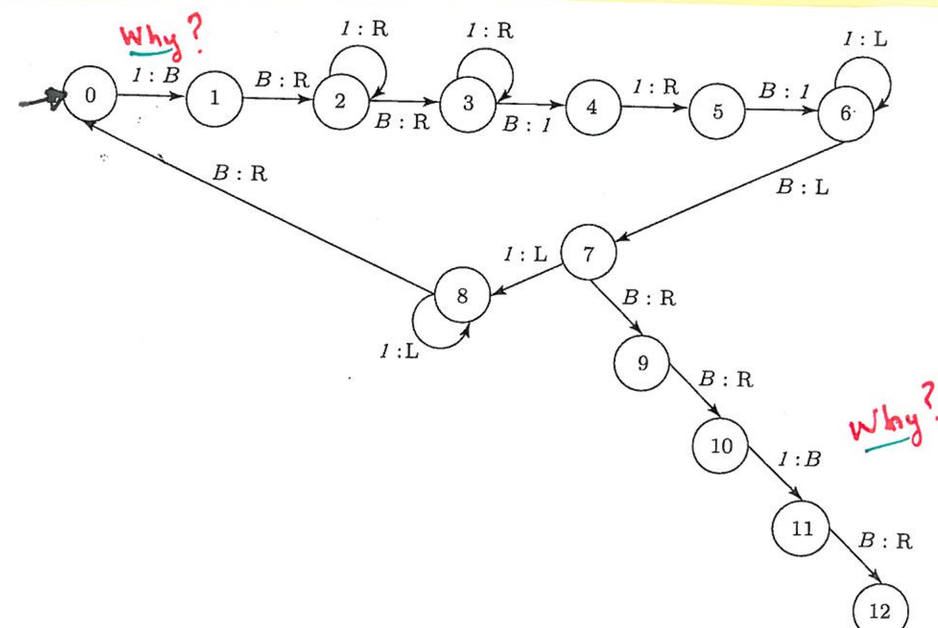


Figure 1.5.2 A Turing Machine That Computes $f(n) = 2n$.

We trace M when $n = 2$

B111B // initial tape

↑

q_0

⋮

B11111B // final tape

↑

q_{12}

Number-Theoretic Functions con't.

- Example 1.5.4 A Turing machine M that computes the addition function

$f(n,m) = n+m$

- Let us trace M when $n=2$ and $m=3$

B 111 B 1111 B //initial tape

go

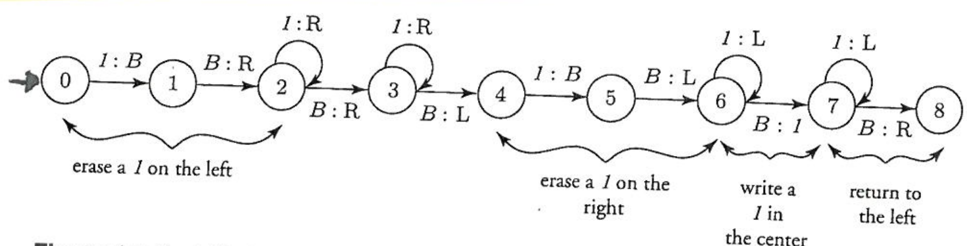


Figure 1.5.3 A Turing Machine That Computes the Binary Addition Function.

- A last word ...

DEFINITION 1.6: *Deterministic Turing machine M computes k-ary partial number-theoretic function f* with $k \geq 1$ provided that:

(i) If M is started scanning the leftmost *1* of an unbroken string of $n_1 + 1$ *1s* followed by a single blank followed by an unbroken string of $n_2 + 1$ *1s* followed by a single blank ... followed by an unbroken string of $n_k + 1$ *1s* on an otherwise blank tape, where function f happens to be defined for arguments $n_1, n_2, \ldots, n_k$, then M halts scanning the leftmost *1* of an unbroken string of $f(n_1, n_2, \ldots, n_k) + 1$ *1s* on an otherwise blank tape.

(ii) If M is started scanning the leftmost *1* of an unbroken string of $n_1 + 1$ *1s* followed by a single blank followed by an unbroken string of $n_2 + 1$ *1s* followed by a single blank ... followed by an unbroken string of $n_k + 1$ *1s* on an otherwise blank tape, where function f is undefined for arguments $n_1, n_2, \ldots, n_k$, then M does not halt scanning the leftmost *1* of an unbroken string of *1s* on an otherwise blank tape.

It should be plain that Example 1.5.4 is strictly in accordance with this definition where $k = 2$ and $f(n, m) = n + m$.

Number Theoretic Functions con't.

- One more last word ...

DEFINITION 1.7: A partial number-theoretic function f is said to be *Turing-computable* if there exists a Turing machine M that computes f in the sense of Definition 1.6.

- and another ...

REMARK 1.5.1: Every Turing machine with input alphabet $\Sigma = \{1\}$ computes some unary partial number-theoretic function.

Finally ... some homework

EXERCISES FOR §1.5

1.5.1 hwk (a) Suppose that Turing machine M, when started scanning the leftmost 1 of an unbroken string of 4 1s on an otherwise blank tape, ultimately halts scanning the leftmost of 10 1s on an otherwise blank tape. Also, M, when started scanning the leftmost 1 of any unbroken string of 5 1s on an otherwise blank tape, ultimately halts scanning the leftmost of 13 1s on an otherwise blank tape. Other input/output pairs are indicated in Table 1.5.1.

What unary number-theoretic function is computed by M in accordance with our conventions?

Table 1.5.1

Number of 1s on Tape Initially	Number of 1s on Tape Ultimately
1	1
2	4
3	7
4	10
5	13
6	16
...	...

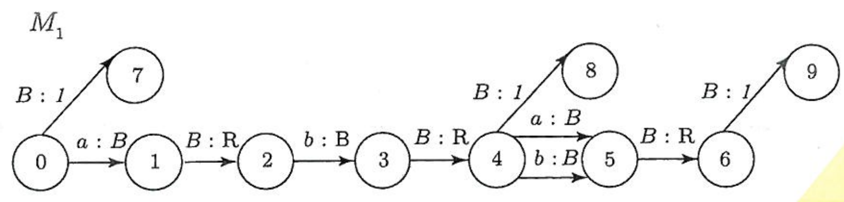
VIII

Homework

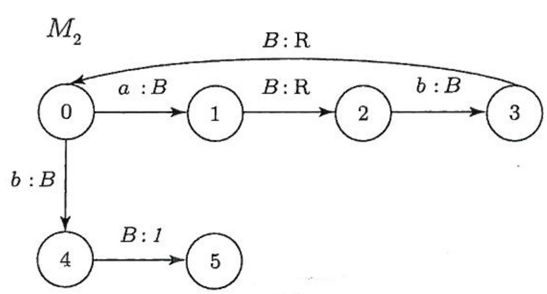
EXERCISES FOR §1.8 (END-OF-CHAPTER REVIEW EXERCISES)

1.8.1. hwk Use set abstraction to describe the language accepted by each Turing machine whose state diagram appears in Figures 1.8.1(a)–(f).

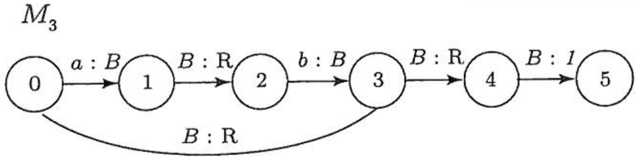
a–c. only



(a)



(b)



(c)

Figure 1.8.1

VIII

Homework con't.

1.8.2. Consider the Turing machine M_1 whose state diagram appears in Figure 1.8.2(a).

(a) What happens if M_1 is started scanning the leftmost of three *1*s on an otherwise blank tape. That is, what is M_1's halting configuration?

(b) What happens if M_1 is started scanning the leftmost of four 1s on an otherwise blank tape. That is, what is M_1's halting configuration?

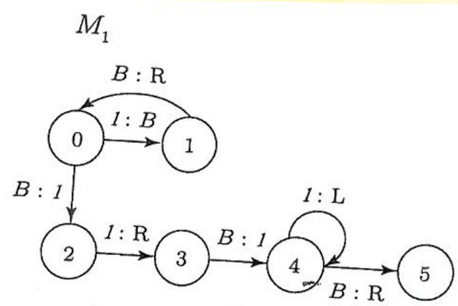
(c) Using your answers to (a) and (b) and observing our adopted conventions regarding representation of numerical input/output, fill in the four blanks below, where f is the number-theoretic function computed by M_1.

$f(\_\_) = \_\_$ (corresponding to (a))

$f(\_\_) = \_\_$ (corresponding to (b))

(d) Finally, using your answers to (c), fill in the blank below so as to characterize the unary function f computed by M_1.

$f(n) = \_\_\_\_$



(a)

Figure 1.8.2

Homework - con't.

1.8.6. hwk (a) Suppose that Turing machine M computes the unary number-theoretic function f defined by

$$f(n) = n^2 + 6n + 3$$

and suppose, further, that M is started scanning the leftmost *1* in an unbroken string of six *1*s on an otherwise blank tape. Then M will halt scanning the leftmost *1* in an unbroken string of how many *1*s?

1.8.8. hwk Present the state diagram of a Turing machine that accepts the language of words over $\{a, b\}$ whose length is 3 or more.

1.8.9. hwk Present the state diagram of a Turing machine that accepts the language of words over $\{a, b\}$ that begin with symbol a and end with symbol b.