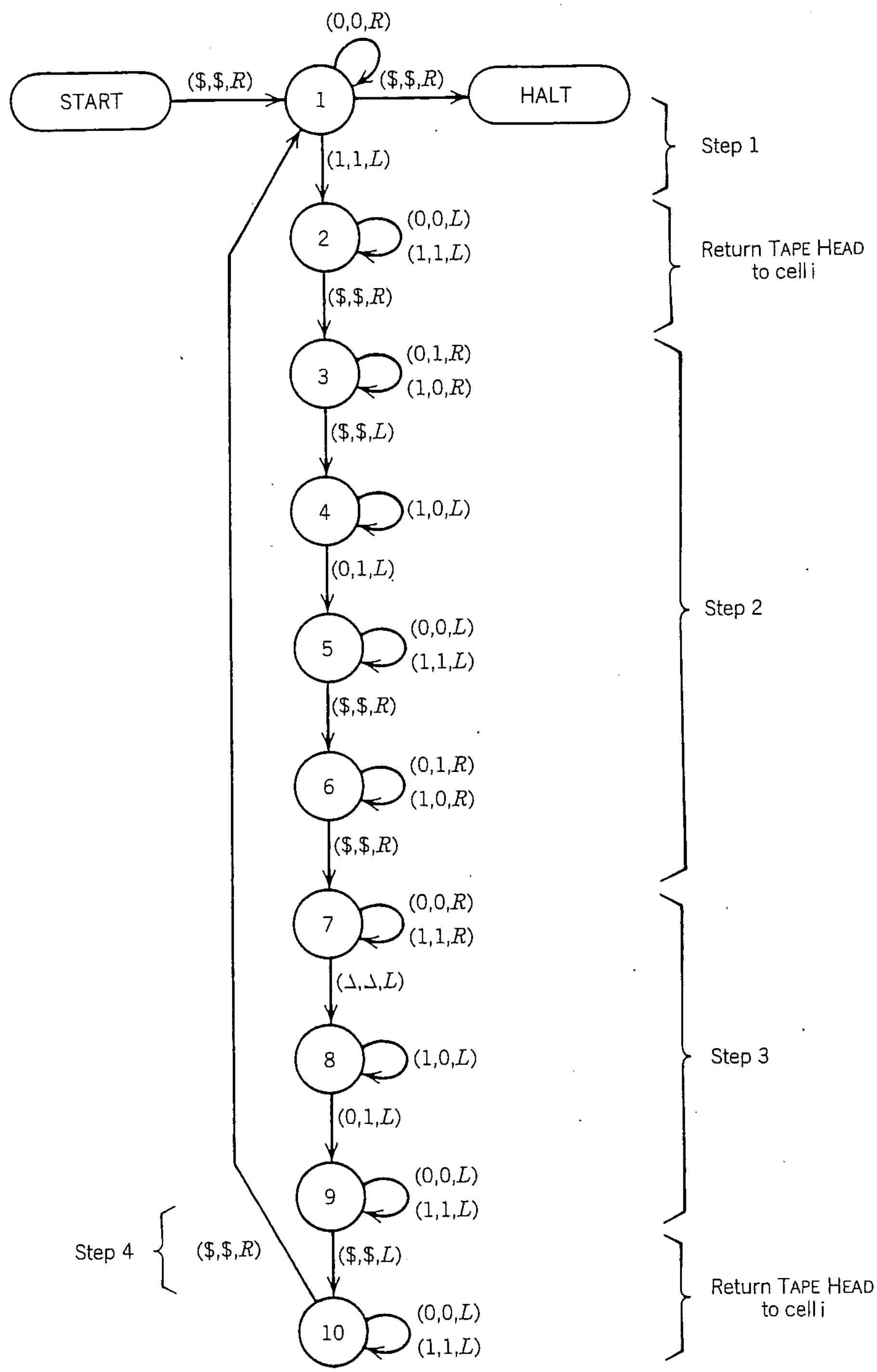

Let us run this machine on the input $10$0110 in an attempt to add 2 and 6 in binary.

START	1		2	3	3
$10$0110	→ $10$0110	(x ≠ 0)	→ $10$0110	→ $10$0110	→ $00$0110
3	4	4	5	6	6
→ $01$0110	→ $01$0110	→ $00$0110	→ $10$0110	→ $10$0110	→ $00$0110
6		7	7	7	7
→ $01$0110	(x ← x − 1)	→ $01$0110	→ $01$0110	→ $01$0110	→ $01$0110
7	8	9	9	9	9
→ $01$0110Δ	→ $01$0110	→ $01$0111	→ $01$0111	→ $01$0111	→ $01$0111
	10	10	10	1	1
(y ← y + 1)	→ $01$0111	→ $01$0111	→ $01$0111	→ $01$0111	→ $01$0111

	2	2	3	3	3
($\mathbf{x} \neq \mathbf{0}$)	→ $\$\underline{0}1\$0111$	→ $\underline{\$}01\$0111$	→ $\$\underline{0}1\$0111$	→ $\$\underline{1}1\$0111$	→ $\$10\underline{\$}0111$
4	5	5	6	6	6
→ $\$1\underline{0}\$0111$	→ $\$\underline{1}1\$0111$	→ $\underline{\$}11\$0111$	→ $\$\underline{1}1\$0111$	→ $\$0\underline{1}\$0111$	→ $\$00\underline{\$}0111$
	7	7	7	7	7
($\mathbf{x} \leftarrow \mathbf{x} - \mathbf{1}$)	→ $\$00\$\underline{0}111$	→ $\$00\$0\underline{1}11$	→ $\$00\$01\underline{1}1$	→ $\$00\$011\underline{1}$	→ $\$00\$0111\underline{\Delta}$
8	8	8	8	9	
→ $\$00\$011\underline{1}$	→ $\$00\$01\underline{1}0$	→ $\$00\$0\underline{1}00$	→ $\$00\$\underline{0}000$	→ $\$00\underline{\$}1000$	($\mathbf{y} \leftarrow \mathbf{y} + \mathbf{1}$)
10	10	10	1	1	1
→ $\$0\underline{0}\$1000$	→ $\$\underline{0}0\$1000$	→ $\underline{\$}00\$1000$	→ $\$\underline{0}0\$1000$	→ $\$0\underline{0}\$1000$	→ $\$00\underline{\$}1000$
	HALT				
($\mathbf{x} = \mathbf{0}$)	→ $\$00\$\underline{1}000$				

The correct binary total is 1000, which is on the TAPE when the TM halts. ■

DEFINITION

If a TM has the property that for every word it accepts, at the time it halts, it leaves one solid string of a's and b's on its TAPE starting in cell i, we call it a **computer.** The input string we call the **input** (or **string of input numbers**), and we identify it as a sequence of nonnegative integers. The string left on the TAPE we call the **output** and identify it also as a sequence of nonnegative integers. ■

In the definition above, we use the semiambiguous word "identify" because we do not wish to restrict ourselves to unary encoding or binary encoding or any other particular system.

COMPUTABLE FUNCTIONS

Now we finally know what a computer is. Those expensive boxes of electronics sold as computers are only approximations to the real McCoy. For one thing, they almost never come with an infinite memory like a true TM. At this stage in our consideration, we are dealing only with zero and the positive integers. Negative numbers and numbers with decimal points can be encoded into nonnegative integers for TMs as they are for electronic digital computers. We do not worry about this generality here. Let us define the new symbol "$\dot{-}$" to use instead of the regular minus sign.

DEFINITION

If m and n are nonnegative integers, then their **simple subtraction** is defined as

$$m \dot{-} n = \begin{cases} m - n & \text{if } m \geqslant n \\ 0 & \text{if } m \leqslant n \end{cases}$$

Essentially what $\dot{-}$ does is perform regular subtraction and then rounds all negative answers back up to 0. ■

Simple subtraction is often called **proper subtraction** or even **monus.**