Algorithmically, the function \( f(x) \) is defined as:

\[
f(x) = \begin{cases} 
1 & \text{if } x \leq 100 \\
0 & \text{otherwise}
\end{cases}
\]

For each of the 100 terms in the sequence, we compute the following:

- Compute the remainder of the term when divided by 10.
- If the remainder is 0, add 1 to the count.
- Otherwise, add 0.

The count is the value of \( f(x) \) for the term at position \( x \).

The diagram illustrates the algorithm for the given example.
EXERCISES

For example, let us compute $A_2^+$ assuming that the top three rows of Figure 6.9 are blank. (We begin by locating $A_2^+$.) The possible right-hand sides in $A_2^+$ are $A$, $AB$, and $BC$. Only the first of these is actually a right side, so we write the corresponding left side $A_2$ to the right of Figure 6.9. Finally, we consider a new right side since $A_2^+$ is not already in $A_2$.

**Figure 6.10** Traversal pattern for computation of $A_2^+$

**Figure 6.9** Table of $A_2^+$

**Figure 6.8** The LALR(1) algorithm is illustrated. The time spent in step (7) is $O(n^2)$, since each set of a production is computed in step (7) and then used in step (11). The correct step to be executed in step (5) is $O(n)$ if it is executed at each execution of the algorithm.

The entire algorithm is iterated until the time spent in step (7) is $O(n^2)$, since the correct time spent in step (7) is $O(n^2)$ if it is executed at each execution of the algorithm.