2.10 PAT Data Structure

- A Continuous text input data structure is indexed in contiguous “n” character tokens using n-grams with interword symbols between processing tokens.
- A continuous text input data structure is addressed differently using PAT trees and PAT arrays.
- The name PAT is short for PAtriciaTrees (PATRICIA stands for Practical Algorithm To Retrieve Information Coded In Alphanumerics.)

- *Sistring (Semi-infinite string)*
  - The input stream is transformed into a searchable data structure consisting of substrings called sistring or semi-finite string.
  - In creation of PAT trees, each position in the input string is the anchor point for a sub-string that starts at that point and includes all new text up to the end of the input.
  - All substrings are unique.
  - A substring can start at any point in the text and can be uniquely indexed by its starting location and length.
  - Substring may go beyond the length of the input stream by adding additional null characters.
  - Figure 4.9 shows some possible sistrings for an input text.

Text	Economics for Warsaw is complex.
sistring 1	Economics for Warsaw is complex.
sistring 2	conomics for Warsaw is complex.
sistring 5	omics for Warsaw is complex.
sistring 10	for Warsaw is complex.
sistring 20	w is complex.
sistring 30	ex.

Figure 4.9 Examples of sistrings

- *PAT tree*
  - is an unbalanced, binary digital tree defined by the sistrings.
  - The individual bits of the sistrings decide the branching patterns with zeros branching left and ones branching right.
  - PAT trees also allow each node in the tree to specify which bit is used to determine the branching via bit position or the number of bits to skip from the parent node.
  - This is useful in skipping over levels that do not require branching.
  - The key values are stored at the leaf nodes (bottom nodes) in the PAT Tree.
  - For a text input of size "n" there are "n" leaf nodes and "n-1" at most higher level nodes.
  - It is possible to place additional constraints on sistrings for the leaf nodes.

INPUT		100110001101
	sistring 1	1001....
	sistring 2	001100...
	sistring 3	01100....
	sistring 4	11.....
	sistring 5	1000...
	sistring 6	000.....
	sistring 7	001101
	sistring 8	01101

Figure 4.10 Sistrings for input "100110001101"

  - Figure 4.10 gives an example of the sistrings used in generating a PAT Tree.
  - If the binary representations of "h" is (100), "o" is (110), "m" is (001) and"e" is (101) then the word "home" produces the input 100110001101.
  - Using the sistrings, the full PAT binary tree is shown in Figure 4.11.

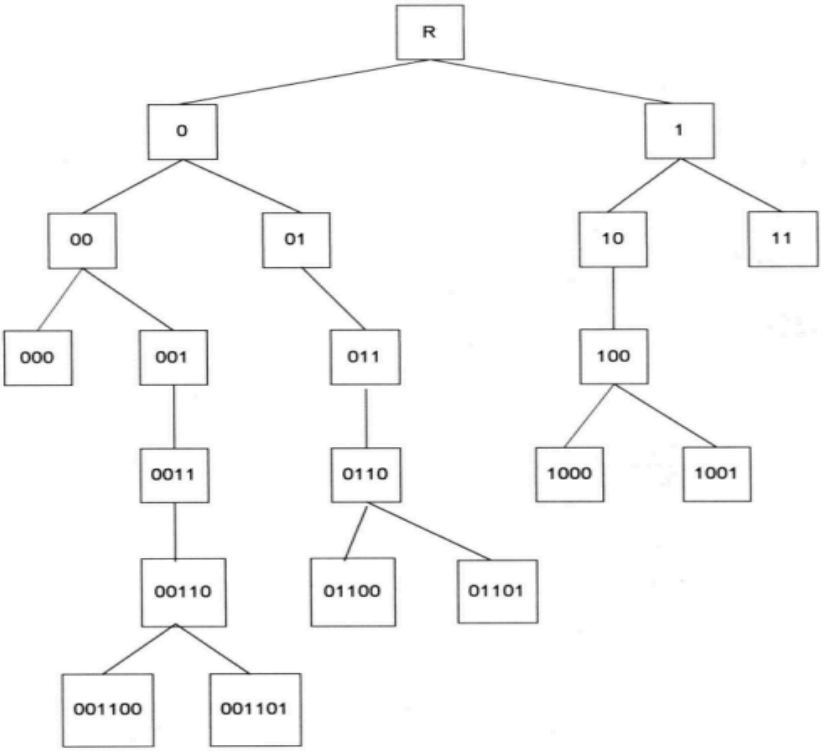


Figure 4.11 PAT Binary Tree for input "100110001101"

_ A more compact tree where skip values are in the intermediate nodes is shown in Figure 4.12.

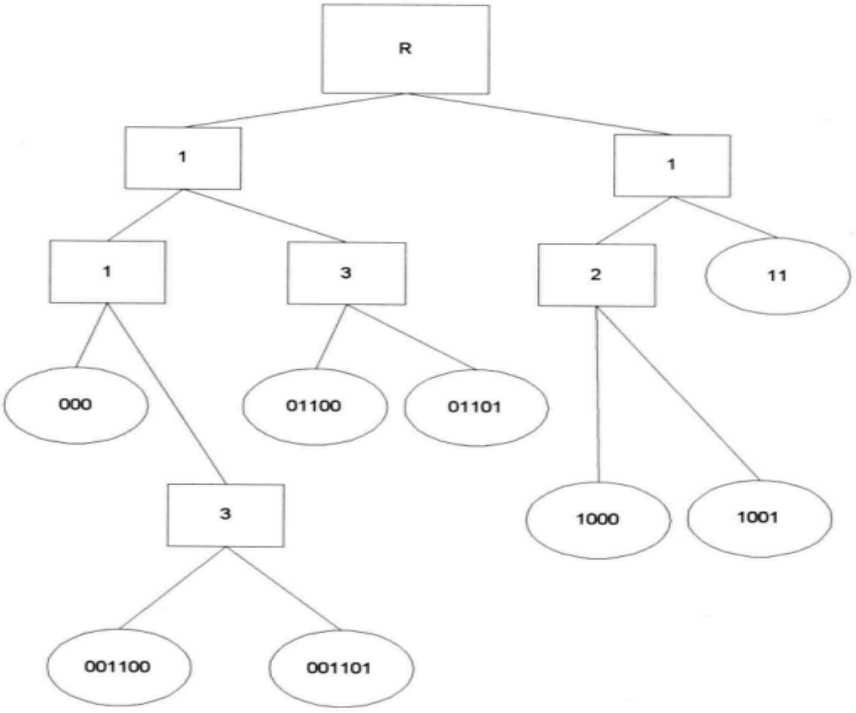


Figure 4.12 PAT Tree skipping bits for "100110001101"

- *Advantages and Disadvantages*
  - PAT trees are ideal for prefix searches.
  - Suffix, imbedded string, and fixed length masked searches are easy if the total input stream is used in defining the PAT tree.
  - Fuzzy searches are very difficult because large number of possible sub-trees could match the search term.
  - PAT arrays have more accuracy than Signature files
  - ability to string searches that are inefficient in inverted files (e.g., suffix searches, approximate string searches, longest repetition).
  - It is not used in any major commercial products.