2.4.5 Formal Definition of Suffix Trees

The reader may use the 'ananas' example from Sect. 1.2.2 as an illustration for the formal concepts introduced here (see Fig. 1.7). Given string T of length n, a *suffix tree* for T is a rooted tree with leaves each labelled with a certain number, edges labelled with non-empty substrings of T in a certain way, additional links between inner nodes called *suffix links*, and a distinguished link to a node called *working position*, such that several properties are fulfilled. To state these properties, we introduce the notion of a *position in the tree* and the *path label* of a position. First, a position is a pointer pointing either to a node of the tree or pointing between two consecutive characters of an edge label. Given a position, its *path label* is the string obtained by concatenating all characters occurring at edges on the path from the root to the position under consideration. As a special case, the *node label* of some node is defined to be the path label of the position pointing to that node. Now, properties of a suffix tree can be defined as follows:

- Every inner node has at least two successor nodes.
- Edges leaving some node are labelled with substrings that have different first characters.
- Every leaf has as node label of some suffix $T[j \ldots n]$ of T and, in that case, is labelled with 'j'.
- Every suffix appears as a path label of a position in the tree.
- Every inner node with path label xw, for a character x and string w, has a so-called *suffix link* starting at that node and pointing to an inner node with path label w in case that w is non-empty, and pointing to the root otherwise.

For more extended trees, it is convenient to present them in a horizontal left-to-right manner. The suffix tree shown in Fig. 1.7 for string $T =$ 'ananas' is special in the sense that all suffixes of T appear as path labels of leaves. The following diagram shows a suffix tree[1] for the string 'mama' (Fig. 2.8).

Observe that two of its suffixes ('ma' and 'a') are represented as path labels of non-leaf positions. The reason for this is that string 'mama' contains suffixes that are at the same time prefixes of other suffixes. This is not the case for the string 'ananas'. By using an additional end marker symbol we can always enforce that suffixes are represented as path labels of leaves. Figure 2.8 shows a suffix tree for string 'mama', as well as for 'mama$'.

[1] In the literature, "suffix tree" for a string T is sometimes understood as suffix tree (in our sense) for string T\$ with additional end marker symbol, whereas a suffix tree (in our sense) is called "implicit suffix tree". Sometimes, suffix links and actual working position are not introduced as explicit components of a suffix tree. Since in the description of Ukkonen's algorithm in Chap. 4 suffix trees for prefixes of a string T\$ play a central role, and such prefixes do not end with end marker \$, we prefer to use the definition of suffix trees as given above.