Hence, for any positive integer m, up to isomorphism there exist exactly $1+\frac{1}{2}(m-1)(n-1)$ m-element n-dimensional commutative quasi-trivial superassociative systems.

THEOREM 11 (classification of a certain class of commutative quasi-trivial superassociative operations of odd arity). *Let n be some even positive integer and let (B, g), $|B|>1$, be some algebra with one $(n+1)$-ary operation. Then t.f.a.e.:*

(i) *(B, g) is an n-dimensional commutative quasi-trivial superassociative system and there hold* (a) *or* (b):

(a) *There exists at most one $x \in B$ such that $g(x, y, \ldots, y) = y$ for any $y \in B$.*

(b) *There exists some $a \in B$ such that $g(a, y, \ldots, y) = y$ for any $y \in B$ and such that there exists some $b \in B$ with $g\big(a(\frac{1}{2}n), b, \ldots, b\big) \neq b$.*

(ii) *There exists some total ordering $\leqslant$ on B, there exists some final segment C of $(B, \leqslant)$ and there exists some integer i, $1 \leqslant i \leqslant \frac{1}{2}n$, such that*

$$g(x_0, \ldots, x_n) = \begin{cases} m_{i,\leqslant}(x_0, \ldots, x_n) & \text{if } (x_0, \ldots, x_n) \in C^{n+1}, \\ m_{1,\leqslant}(x_0, \ldots, x_n) & \text{otherwise} \end{cases}$$

$(x_0, \ldots, x_n \in B)$.

Remark. The following example shows that there exist n-dimensional commutative quasi-trivial superassociative systems, n even, neither satisfying (i) (a) nor (i) (b): Put $B := \{0, 1, 2, 3\}$, $n := 2$, $g(x, x, y) = g(x, y, x) = g(y, x, x) := x$ for any $x, y \in B$ and $g(x, y, z) \equiv -(x+y+z) \bmod 4$ for any three mutually distinct elements $x, y, z \in B$.

References

[1] R. M. Dicker, *The substitutive law*, Proc. London Math. Soc. 3. Ser. 13 (1963), pp. 493–510.

[2] H. Lausch and W. Nöbauer, *Algebra of Polynomials*, Amsterdam 1973.

[3] K. Menger, *Algebra of analysis*, Notre Dame Math. Lect. 3 (1944).

[4] — *Superassociative systems and logical functors*, Math. Ann. 157 (1964), pp. 278–295.

[5] H. Skala, *Irreducibly generated algebras*, Fund. Math. 67 (1970), pp. 31–37.

TECHNISCHE UNIVERSITÄT WIEN
INSTITUT FÜR ALGEBRA UND MATHEMATISCHE STRUKTURTHEORIE
Vienna

Accepté par la Rédaction le 27. 2. 1978

On vertices and edges in maximum path-factors of a tree

by

Zdzisław Skupień and **Władysław Zygmunt** (Kraków)

Abstract. The paper presents proofs for part of the results announced in [11]. It develops a method of classifying the edges and the vertices of a tree T with respect to their appearance in maximum path-factors of T.

1. Introduction. Since Ore's pioneering work [7] in 1961, different publications concerning Hamiltonian graphs have dealt with the covering of vertices by (or partition of vertices into) disjoint (possibly trivial) paths in an ordinary graph, say G. Most of these papers deal with the invariant of G introduced by Barnette [1]. Following Skupień [8] we will denote this invariant by $\pi_0(G)$, and call it the vertex-path partition number of G, where $\pi_0(G)$ is the minimum number of paths among the path partitions of vertices of G.

Recently, new related invariants, namely Hamiltonian completion number $\mathrm{hc}(G)$ and Hamiltonian shortage $\mathrm{s_H}(G)$, have been independently introduced by Goodman and Hedetniemi [3], and Skupień [8], [9]. In general, these new invariants coincide. Namely, both equal 0 when G is Hamiltonian, and both equal $\pi_0(G)$ when G is a non-trivial non-Hamiltonian graph. Only for $G = K_1$ we have $\pi_0(K_1) = \mathrm{hc}(K_1) = \mathrm{s_H}(K_1)-1 = 1$.

In a series of papers sufficient conditions have been found for either $\pi_0(G) \leqslant s$ or $\mathrm{s_H}(G) \leqslant s$, where s is an integer.

The problem of determining $\pi_0(G)$ or $\mathrm{s_H}(G)$ is considered independently in [2], [3], and [8]. In each of these papers algorithms for determining $\pi_0(G)$ in the case where G is a tree or forest are developed. Algorithms presented in [2] and [3] are very similar to each other. Two other algorithms, based on labelling the vertices of a tree, are presented in [8].

Evaluating π_0 for trees is of special importance. Namely, in [2] and [3] it is noted that, for a connected graph G,

$$\pi_0(G) = \min\{\pi_0(T)\colon\ T \text{ is spanning tree of } G\}.$$

In general (cf. [10]),

$$\pi_0(G) = \min\{\pi_0(F)\colon\ F \text{ is a spanning forest of } G, \text{ with components which are spanning trees of components of } G\}.$$

In [2] and [3] it is also noted that $\pi_0(G)$ or $\mathrm{hc}(G)$ can be introduced in an information retrieval context, not only in connection with Hamiltonian graph theory. As a matter of fact, those practical applications show that determining partitions of vertuces of a non-Hamiltonian graph G into $\pi_0(G)$ path is more important than determining $\pi_0(G)$ itself. It is worthy of notice in this context that problem of determining all those partitions if G is a non-Hamiltonian graph is stated as Problem 3 in [8]. We shall deal with this problem in the case where G is a tree (or a forest), though in this case the problem is solved in Kania [4] by using linear programming. The present paper is designed to yield a base for an alternative combinatorial solution of the problem in the case of a tree. This solution will be presented in [13].

In what follows, we consider an equivalent to the above-mentioned path covering or path partition of vertices of G. It is called a path-factor of G. By a path-factor of G we mean a factor whose each component is a path (possibly trivial). A path-factor is a maximum path-factor if its size is the largest possible. Though a path-factor is not a subgraph with prescribed vertices, its name and its structure resemble the notion of a 1-factor or a k-factor where k is a vector. Also our investigations of path-factors, presented below, resemble similar investigations as regards classifying edges with respect to their appearance in k-factors (cf. Lovász [5] and [6]). However, we shall classify not only edges but also vertices of a tree with respect to their appearance in path-factors of a tree.

The results of this paper were announced in [11] and [10], where procedures for determining all maximal path-factors are sketched. In another related paper [12] we show that decomposing a tree with a vertex of degree 2 can simplify procedures for determining path-factors.

2. Basic terminology and notation. We shall use the common terminology and notation of graph theory with special additions which were introduced in [8], [11] and [12]. For the sake of completeness we recall some definitions and notation.

Only ordinary (i.e., simple) graphs G will be considered. The terms *factor* and *spanning subgraph* will be identified.

Given a graph $G = (V, E)$ and either a set of vertices V_1 or another graph G^1 with the vertex set V_1, each of the symbols $G\text{-}V_1$ and $G\text{-}G^1$ denotes the subgraph of G induced by the subset of vertices $V\text{-}V_1$. Given a set of edges E_1, the symbol $G\text{-}E_1$ denotes the factor of G with the edge set $E\text{-}E_1$. The *number of components* of G is denoted by $k(G)$.

By a *path* P_n with $n \geqslant 1$ we mean a graph with n vertices $x_1, x_2, \ldots, x_n$ and $n-1$ edges $x_i x_{i+1}$ $(i = 1, 2, \ldots, n-1)$. For P_n the notation

$$P_n = [x_1, x_2, \ldots, x_n]$$

is used and the vertices x_1 and x_n are called the *end-vertices* of P_n. Moreover, $P_1 = [x_1]$ with $E(P_1) = \emptyset$ is called a *trivial path*.

The *degree* of a vertex x in G is denoted by $d(x, G)$. A vertex of degree 1 or 0 in G is called a *hanging vertex*. Each of the remaining vertices is called the *inner vertex* and any vertex of degree greater than 2 is said to be a *branching vertex*.

By a *string* in G we mean a non-trivial path whose each inner vertex (if any) is of degree 2 in G and each of the end-vertices is either hanging or branching in G. A string is a *hanging string* if (at least) one of its end-vertices is hanging. A string P in G is said to be *attached to a vertex* x if x is an end-vertex of P and is a branching vertex in G. Then the edge of P, incident to x is called the *edge of attachment* of P.

A branching vertex v is a *hanging branching vertex* if all or all but one of the strings attached to v are hanging. An edge, say e_v, which is incident to a hanging branching vertex v and does not belong to any hanging string is called the *edge of attachment of the hanging branching vertex* v.

It is clear that each tree with two or more branching vertices has at least two hanging branching vertices each of which has an edge of attachment.

By a *path-factor* of G we mean a factor each component of which is a path (possibly trivial). A path-factor of G is called a *maximum path-factor* if its size is maximal among all path-factors of G. Let $\mathscr{W}(G)$ and $\mathscr{S}(G)$ denote the collections of all path-factors and all maximum path-factors of G, respectively. The *vertex-path partition number* $\pi_0(G)$ of G is defined as follows:

$$\pi_0(G) = \min\{k(W)\colon\ W \in \mathscr{W}(G)\}.$$

Observe that a path-factor is maximum if the number of its components is the smallest possible; more precisely,

$$W \in \mathscr{S}(G) \Leftrightarrow W \in \mathscr{W}(G) \wedge k(W) = \pi_0(G).$$

An edge is called *compulsory* or *forbidden* in G if it belongs to all or to no maximum path-factors of G, respectively; otherwise the edge is called *free* in G. A path in G is said to be a *compulsory* (*green*) *path* if it is a non-trivial path whose each edge is compulsory [green] in G.

The plucking operation $T \mapsto \mathrm{pl}(T)$ on trees T and the ith derived forest $F_i(T)$ of a tree (or a forest) T, which are concepts of great importance in what follows, are introduced in the next section.

3. Some general properties of maximum path-factors. We start with stating a simple result on maximum path-factors in a disconnected graph.

(3.1) THEOREM. *Let graphs G^i be components of a graph G, $i = 1, 2, \ldots, k(G)$. Then a path-factor S of G is maximum iff, for each G^i, the intersection $S \cap G^i$ is a maximum path-factor of G^i, i.e.,*

$$S \in \mathscr{S}(G) \Leftrightarrow S \in \mathscr{W}(G) \quad \text{and} \quad \forall i\colon\ 1 \leqslant i \leqslant k(G) \Rightarrow S \cap G^i \in \mathscr{S}(G^i).$$

The proof follows from the following three obvious formulas:

$$\pi_0(G) = \sum_{i=1}^{k(G)} \pi_0(G^i),$$

$$S \in \mathscr{W}(G) \Rightarrow k(S) = \sum_{i=1}^{k(G)} k(S \cap G^i),$$

$$k(S \cap G^i) \geqslant \pi_0(G^i) \quad \text{with } i = 1, 2, \ldots, k(G),\ S \in \mathscr{W}(G).\ \blacksquare$$

Since, for $S \in \mathscr{W}(G)$,

$$S = \bigcup_{i=1}^{k(G)} (S \cap G^i),$$

we have the following

(3.2) Corollary. *If G^i are components of G, then*

$$\mathscr{S}(G) = \{\bigcup_{i=1}^{k(G)} S^i\colon\ S^i \in \mathscr{S}(G^i)\}.$$ ■

(3.3) Corollary. *Each edge of any path which is a component of the graph G is compulsory in G.* ■

(3.4) Definition. Let $F_1(T)$ denote the *forest derived from* (or the *derived forest* of) a tree (or, more generally, a forest) T, i.e., $F_1(T)$ stands for a forest obtained by deleting the edges of attachment of all hanging branching vertices in T.

Since

$$\pi_0(F_1(T)) = \pi_0(T)$$

(cf. the formula (13) in [8]), each maximum path-factor of $F_1(T)$ is also a maximum path-factor of T, that is,

(3.5) $$\mathscr{S}(F_1(T)) \subseteq \mathscr{S}(T).$$

In what follows we shall make use of the so-called *plucking operation* pl on trees. This operation was used in [8] without giving it a name.

(3.6) Definition. Given a tree T. Let $\mathrm{pl}(T)$ denote either

(a) the tree T if T does not contain any branching vertex or

(b) the subtree (possibly the empty graph K_0) obtained from T by deleting all hanging branching vertices together with incident edges and all hanging strings attached to those vertices.

(3.7) Remark. Either $\mathrm{pl}(T)$ is a component of the derived forest $F_1(T)$ or $\mathrm{pl}(T) = K_0$. ■

Let pl^k denote the *k-th iterate* of the operation pl, $k \geqslant 1$, and let pl^0 be the identity operation on T. Since the tree T is finite, there is an integer h (e.g., $h = |V(T)|$) such that

$$\mathrm{pl}^h(T) = \mathrm{pl}^{h+1}(T) = \ldots =: \mathrm{pl}^\infty(T).$$

Note that $\mathrm{pl}^\infty(T)$ is either a path (possibly trivial K_1) or the empty graph K_0.

Analogously, let $F_0(T) = T$ and let $F_i(T)$ denote the *i-th derived forest* of T, i.e., the forest derived from the $(i-1)$-th derived forest $F_{i-1}(T)$, $i = 1, 2, \ldots$ Obviously, there exists an integer j such that

$$F_j(T) = F_{j+1}(T) = \ldots = F_\infty(T).$$

Hence

(3.8) Remark. $F_\infty(T)$ is clearly a spanning forest of T each component of which has at most one branching vertex. $F_\infty(T)$ is called the *simplest derived forest* of T. ■

The following observation generalizes Remark (3.7):

(3.9) Remark. Either $\mathrm{pl}^k(T)$ is a component of the kth derived forest $F_k(T)$ or $\mathrm{pl}^k(T) = K_0$, $k = 1, 2, \ldots$ ■

Iterating formula (3.5) gives

(3.10) $$\mathscr{S}(F_k(T)) \subseteq \mathscr{S}(T), \quad k = 1, 2, \ldots$$

Hence, by the definition of the derived forests, we have

(3.11) Corollary. *No edge of attachment of a hanging branching vertex in any $\mathrm{pl}^k(T)$, $k \geqslant 0$, is compulsory in T.* ■

Therefore, by Remark (3.7) and Corollary (3.2), all components of any maximum path-factor of $\mathrm{pl}(T)$ are components of a certain maximum path-factor of T. More generally, we have

(3.12) Corollary. *For any maximum path-factor ${}^kS$ of $\mathrm{pl}^k(T)$, there is a certain maximum path-factor S of T such that each component of ${}^kS$ is a component of S.* ■

Consider any maximum path-factor S of T. It was noted in [8] p. 490, that edges of attachment of hanging branching vertices can be eliminated from the maximum path-factor S of T. Namely, if S contains edges of attachment of hanging branching vertices of T, then there exists an $\tilde{S} \in \mathscr{S}(F_1(T))$ ($F_1(T)$ being the forest derived from T) such that the intersection of $\tilde{S}$ and $\mathrm{pl}(T)$ is identical with that of S and $\mathrm{pl}(T)$. In general, for any $k \geqslant 1$, we have

(3.13) $$\forall S \in \mathscr{S}(T)\, \exists \tilde{S} \in \mathscr{S}(F_k(T))\colon\ S \cap \mathrm{pl}^k(T) = \tilde{S} \cap \mathrm{pl}^k(T).$$

So, owing to Remark (3.9) and Theorem (3.1), we have the following

(3.14) Corollary. *Given any maximum path-factor S of T, $S \cap \mathrm{pl}(T)$ is a maximum path-factor of $\mathrm{pl}(T)$; more generally,*

$$S \in \mathscr{S}(T) \Rightarrow S \cap \mathrm{pl}^k(T) \in \mathscr{S}(\mathrm{pl}^k(T)), \quad k = 1, 2, \ldots$$ ■

(3.15) Theorem. *The collection $\mathscr{S}(T)$ of all maximum path-factors of T induces the collection $\mathscr{S}(\mathrm{pl}^k(T))$ of all maximum path-factors of $\mathrm{pl}^k(T)$, that is,*

$$\mathscr{S}(\mathrm{pl}^k(T)) = \mathscr{S}_k := \{S \cap \mathrm{pl}^k(T)\colon\ S \in \mathscr{S}(T)\}, \quad k = 1, 2, \ldots$$

Proof. The inclusion $\mathscr{S}_k \subseteq \mathscr{S}(\mathrm{pl}^k(T))$ follows from Corollary (3.14). The converse inclusion follows from Corollary (3.12). ■

(3.16) Remark. Any edge which is incident to an inner vertex of a compulsory path and does not belong to that path is clearly forbidden. ■

Now, Theorem (3.15) implies the following

(3.17) Corollary. *An edge of $\mathrm{pl}^k(T)$ is compulsory (forbidden or free) in $\mathrm{pl}^k(T)$, for any $k = 1, 2, \ldots$, iff it is so in T.* ■

Hence, we have the following

(3.18) Corollary. *If an edge of a hanging string P in $\mathrm{pl}^k(T)$, $k \geqslant 0$, is not any edge of attachment of P, then the edge e is compulsory in T; in particular, if $\mathrm{pl}^\infty(T)$ is non-empty and non-trivial then $\mathrm{pl}^\infty(T)$ is a compulsory path in T.* ■

4. Labelling procedure. The following procedure of labelling the vertices of a tree is a modification of one of the procedures presented in [8].

PROCEDURE 1 (of labelling the vertices of a tree T).

1. Associate label 1 with every hanging vertex.

2. Associate label 2 with every hanging branching vertex.

3. Associate label 21 or 3 with each remaining vertex x of any hanging string attached to a vertex with label 2 according as the vertex x is adjacent to a vertex with label 2 or not.

4. Apply the plucking operation to the tree in question.

5. In step 4 one obtains one of the following three graphs:

(a) the empty graph K_0,

(b) a path P (possibly $P = K_1$),

(c) a tree with a branching vertex.

5a. Stop in case (a).

5b. In case (b), for each vertex x which has no label, associate label 12 or 3 with x according as the vertex is hanging or inner in P.

5c. In case (c) associate label 12 with each hanging vertex which has no label and go to step 2.

6. Determine $h := \min\{i\colon \mathrm{pl}^i(T) = \mathrm{pl}^{i+1}(T) (= \mathrm{pl}^{\infty}(T))\}$. (Note that h is the number of transitions to Step 2 while executing Procedure 1 for the tree T).

The above procedure is clearly finite and associates a label with each vertex of the tree T.

The formula for the vertex-path partition number $\pi_0(T)$ of T (cf. formula (12) in [8]) now takes the form:

$$\pi_0(T) = N_1 + N_{12} - N_2 - \varepsilon(\mathrm{pl}^{\infty}(T)) \tag{4.1}$$

where N_i is the number of vertices with label i in T ($i = 1, 2, 12$) and

$$\varepsilon(\mathrm{pl}^{\infty}(T)) = \begin{cases} 1 & \text{if } \mathrm{pl}^{\infty}(T) \text{ is a nontrivial path}, \\ 0 & \text{if } \mathrm{pl}^{\infty}(T) \text{ is } K_1 \text{ or } K_0 . \end{cases}$$

(4.2) DEFINITION. An edge whose end-vertices have labels i and j is said to be the *edge of type* (i-j).

Now since each vertex with label 3 belongs to a hanging string in a certain $\mathrm{pl}^k(T)$ and is not incident to the edge of attachment of this string, Corollary (3.18) implies the following

(4.3) COROLLARY. *Each vertex with label* 3 *is incident to exactly two compulsory edges of which neither is of type* (3-2). ■

Hence we have

(4.4) COROLLARY. *No vertex with label* 3 *is a hanging vertex of a maximum path-factor of* T. ■

The following remark is obvious.

(4.5) Remark. For any vertex x with label 12 there is exactly one incident edge e which is not the edge of attachment of any hanging branching vertex in any $\mathrm{pl}^k(T)$, $k \geqslant 0$, unless x is the unique vertex of $\mathrm{pl}^{\infty}(T)$ (then $\mathrm{pl}^{\infty}(T) = K_1$ and there is no such edge e). ■

Observe that each hanging branching vertex has label 2. Hence, by Definition (3.4) of the derived forest, we have

(4.6) Remark. Each edge not belonging to $F_{\infty}(T)$ is of type (2-j) with possibly $j = 12, 2, 21, 3$ ($j \neq 1$) and if an edge of type (2-i) belongs to $F_{\infty}(T)$ then $i = 1, 12, 21$ ($i \neq 2$, $i \neq 3$). Moreover, each but possibly one of the components of $F_{\infty}(T)$ has exactly one vertex with label 2. Only $\mathrm{pl}^{\infty}(T)$ has no such vertex. (In particular, each branching vertex in $F_{\infty}(T)$ has label 2.) Furthermore, each hanging vertex of $F_k(T)$ ($k \geqslant 0$) has label 1 or 12. ■

The following remark is obvious.

(4.7) Remark. A vertex v with label 21 is of degree 2 in a certain $\mathrm{pl}^k(T)$ and in $F_{\infty}(T)$. ■

Note that neither of the two neighbours of that v in $F_{\infty}(T)$ has label 21. Owing to Remark (4.6), each remaining neighbour of v in T has label 2. Hence we have

(4.8) Remark. An edge of type (21-21) cannot appear in any labelled tree; edges of other types can appear (see Fig. 1). ■

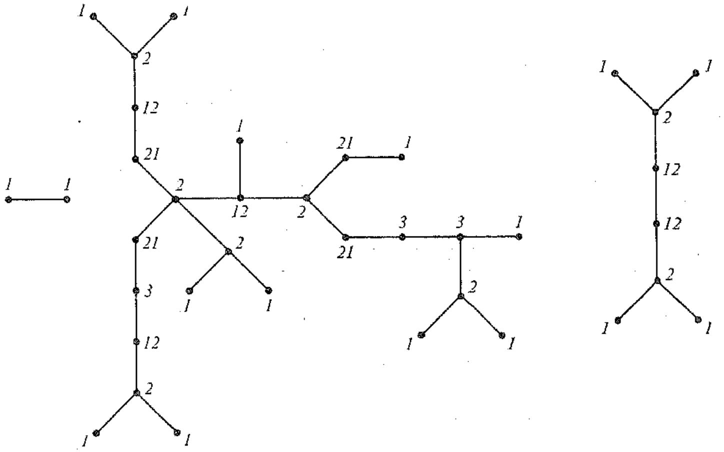

Fig. 1. The labelled forest which contains edges of all possible types (except (21-21)).

(4.9) LEMMA. *Each of the hanging vertices in any maximum path-factor of the tree* T *has label* 1, 12, *or* 21.

Proof. By Corollary (4.4) it suffices to show that each vertex with label 2 is an inner vertex in any maximum path-factor of T. To this end, consider a vertex x with label 2. So the vertex x is a hanging branching vertex of a certain subtree $\mathrm{pl}^k(T)$ of T. Therefore x cannot be a hanging vertex in any path-factor ${}^kS \in \mathscr{S}(\mathrm{pl}^k(T))$. Therefore, using Corollary (3.14), we deduce that Lemma (4.9) is true. ■

Hence, by Corollary (3.12), we have the following

(4.10) COROLLARY. *All hanging vertices in any maximum path-factor of* $\mathrm{pl}^k(T)$, $k\geq 0$, *have labels* 1, 12, *or* 21. ■

On the other hand, since each vertex labelled 1 or 12 is hanging in a certain $\mathrm{pl}^k(T)$, $k\geq 0$, Corollary (3.12) implies

(4.11) COROLLARY. *Each vertex with label* 1 *or* 12 *is hanging in a certain maximum path-factor of* T. ■

(4.12) LEMMA. *Edges of types* (2-2) *and* (2-3) *are forbidden in* T.

Proof. According to Corollary (4.3), no edge of type (2-3) belongs to any maximum path-factor of T. Now suppose that there is a maximum path-factor S of T containing an edge, say xy, of type (2-2). So there is an integer l, $l\geq 0$, such that either

(i) both the vertices x, y belong to $\mathrm{pl}^l(T)$ and neither of them belongs to $\mathrm{pl}^{l+1}(T)$ or

(ii) both the vertices x and y belong to $\mathrm{pl}^l(T)$ and only one of them, say x, belongs to $\mathrm{pl}^{l+1}(T)$.

In case (i) x and y are the only hanging branching vertices of the tree $\mathrm{pl}^l(T)$ (cf. Fig. 2). Then $\mathrm{pl}^{l+1}(T) = K_0$ and, by (4.1),

$$\pi_0(\mathrm{pl}^l(T)) = d(x, \mathrm{pl}^l(T))+d(y, \mathrm{pl}^l(T))-4\,.$$

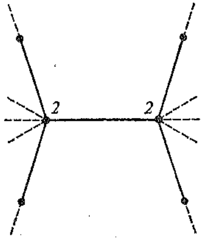

Fig. 2. The tree $\mathrm{pl}^l(T)$

Now observe that if

$$W := S\cap \mathrm{pl}^l(T)$$

then $xy\in E(W)$ and, by Corollary (3.14),

$$\pi_0(\mathrm{pl}^l(T)) = k(W)\geq \big(d(x, \mathrm{pl}^l(T))-2\big)+\big(d(y, \mathrm{pl}^l(T))-2\big)+1 = \pi_0(\mathrm{pl}^l(T))+1$$

— a contradiction.

In case (ii) the vertex x is a hanging branching vertex in a certain tree $\mathrm{pl}^k(T)$ with $k\geq l+1$. According to Corollary (3.14), the graph $S\cap \mathrm{pl}^k(T)$ is a maximum path-factor of $\mathrm{pl}^k(T)$. Furthermore, the vertex x with label 2 is a hanging vertex in $S\cap \mathrm{pl}^k(T)$ — a contradiction of Corollary (4.10). Thus the proof is completed. ■

(4.13) LEMMA. *Let x be a hanging branching vertex of degree* 3 *in some* $\mathrm{pl}^k(T)$ *with* $k\geq 0$, $\mathrm{pl}^k(T)$ *containing at least two hanging branching vertices, and let the edge* e_x *of attachment of x be forbidden. Then if e is an edge of a hanging string in* $\mathrm{pl}^k(T)$, *attached to x then e is compulsory in T.*

Proof. Since, by Corollary (3.17), the edge e_x is forbidden in $\mathrm{pl}^k(T)$, e belongs to every maximum path-factor of $\mathrm{pl}^k(T)$. Hence, by Corollary (3.17), Lemma (4,13) follows. ■

5. Colouring procedure. Recall that in Step 6 of Procedure 1 the following number is determined:

$$h = \min\{i\colon\ \mathrm{pl}^i(T) = \mathrm{pl}^\infty(T)\}\,. \tag{5.1}$$

PROCEDURE 2 (of colouring some edges of a tree T with vertices labelled by Procedure 1).

1. Associate the green colour with every edge of $\mathrm{pl}^h(T)$ and stop if $h = 0$.
2. In $\mathrm{pl}^i(T)-E(\mathrm{pl}^{i+1}(T))$, for successive decreasing values of i ($i = h-1, h-2, \ldots, 1, 0$);

2a. Associate the red colour with each edge of type (2-2) or (2-3) and with each non-coloured one which is incident to an inner vertex of a green path in $\mathrm{pl}^{i+1}(T)$;

2b. Associate the green colour with each edge e which belongs to a hanging string, P, in $\mathrm{pl}^i(T)$ and is not any edge of attachment of P or belongs to a path which is a component of that factor of $\mathrm{pl}^i(T)$ which is obtained by deleting all red edges.

Observe that, for any i ($0\leq i\leq h-1$), each component of the graph $\mathrm{pl}^i(T)-\mathrm{pl}^{i+1}(T)$ is a component of the forest $F_\infty(T)$ (cf. Definition (3.4) of the derived forest). Consequently, in the graph $\mathrm{pl}^i(T)-E(\mathrm{pl}^{i+1}(T))$ (which is considered in Step 2 above) no edge incident to a vertex of $\mathrm{pl}^{i+1}(T)$ belongs to $F_\infty(T)$. Moreover, since, by Remark (4.6), no edge of type (2-2) or (2-3) belongs to $F_\infty(T)$, the following remark is true.

(5.2) Remark. No red edge belongs to the simplest derived forest $F_\infty(T)$ of the tree T, that is, each red edge is the edge of attachment of a hanging branching vertex in a certain subtree $\mathrm{pl}^i(T)$ of T. ■

Note that the green colour is not associated with any edge of attachment of a hanging branching vertex. Hence

(5.3) Remark. Each green edge belongs to $F_\infty(T)$. ■

According to the colouring procedure above each edge of $\mathrm{pl}^i(T)-E(\mathrm{pl}^{i+1}(T))$ which remains non-coloured is the edge of attachment of either a hanging string or a hanging branching vertex in $\mathrm{pl}^i(T)$. Therefore every such edge is incident to a vertex with label 2. Hence, by Remark (4.6) and the fact that edges of types (2-2) and (2-3) are red, we have:

(5.4) Remark. One of the end-vertices of any non-coloured edge has label 2 and the other one has label 1, 12, or 21. ■

The following remark is obvious:

(5.5) Remark. All edges incident to a vertex with label 3 are coloured and exactly two of them are green. ■

Let x be a hanging branching vertex in a certain $\mathrm{pl}^i(T)$ and let e_x be the edge of attachment of x. Now consider the case that $d(x, \mathrm{pl}^i(T)) = 3$ and the edge e_x is red. Then the two remaining edges incident to x in $\mathrm{pl}^i(T)$ are green (cf. Step 2b of Procedure 2). Each of the other edges of T (if any) which are incident to x is of type (2-2) and therefore is red (cf. Step 2a of Procedure 2). In the opposite case either $d(x, \mathrm{pl}^i(T)) \geqslant 4$ or e_x is non-coloured and therefore there are in $\mathrm{pl}^i(T)$ at least 3 non-coloured edges incident to x: we have

(5.6) Remark. Given a vertex x with label 2, either there are at least three non-coloured edges incident to x or all edges incident to x are coloured and exactly two of them are green. ■

(5.7) Remark. Among the edges incident to a vertex with label 21 either there are exactly two green edges and the remaining ones are red or there is exactly one green edge and the remaining ones are non-coloured. ■

Now Remark (4.7) can be replaced by the following one:

(5.8) Remark. For any vertex v with label 21 there is a subtree $\mathrm{pl}^k(T)$ such that the degree $d(v, \mathrm{pl}^k(T)) = 2$ and $\mathrm{pl}^k(T)$ contains all green edges of T incident to v. ■

(5.9) Remark. The edges incident to a vertex with label 12 are all non-coloured except possibly one which is green (cf. Remark (4.5)). ■

(5.10) Remark. The unique edge incident to any vertex with label 1 is either green or non-coloured. ■

We add also three further remarks, which will be used in the next paper [13].

(5.11) Remark. Deleting red edges from T does not result in the appearance of any new hanging vertex. ■

This follows from Remarks (5.5)–(5.10).

Let T^g stand for the green factor of T, that is, let T^g be a factor of T which contains all the green and only the green edges of T. Since any vertex of T is incident to at most two green edges, we have

(5.12) Remark. Each component of the green factor T^g of T is a path (possibly trivial). ■

(5.13) Remark. Let P' be a component of the green factor T^g of T. If one of the end-vertices of P' has label 21, then the remaining one has label 1 or 12 and all the inner vertices (if any) have labels 3. ■

Remark (5.13) follows from Procedures 1 and 2 and Remarks (5.7)–(5.10).

6. Classifications of edges and vertices. Note that Procedure 1 associates one of five labels 1, 2, 3, 12, and 21 to each vertex of a tree T. Thus performing Procedure 1 gives the partition of the vertex set of T into five or fewer classes. Analogously, performing Procedure 2 gives the partition of the edge set of T into three or fewer classes which consist of green, red, or non-coloured edges, respectively. Now we shall show that those two partitions make it possible to classify edges and vertices of T with respect to their appearance in all maximum path-factors of T. Certain auxiliary results have been stated in the preceding sections.

Using descending induction on i where $i = h, h-1, \ldots, 1, 0$ and making use of Corollaries (3.17) and (3.18), Lemmas (4.13) and (4.12), and Remark (3.16), we can prove the following

(6.1) Lemma. *Each green edge is compulsory in* $\mathrm{pl}^i(T)$ *and each red one is forbidden in* $\mathrm{pl}^i(T)$, $i = 0, 1, \ldots, h$. *(Note that (for* $i = 0$*)* $\mathrm{pl}^0(T) = T$.*)* ■

The following obvious lemma is complementary to Lemma (4.9):

(6.2) Lemma. *If each non-coloured edge of a tree is free, then any vertex with label either* 1 *or* 12 *and any one with label* 21 *and with exactly one green edge incident to it is an end-vertex of a certain maximum path-factor of the tree.*

If T denotes a given tree then, according to Corollary (3.12), it suffices to prove Lemma (6.2) for that subtree $\mathrm{pl}^k(T)$ in which the vertex in question belongs to a hanging string. However, then the lemma is obvious.

(6.3) Lemma. *Each edge which remains non-coloured after performing Procedures* 1 *and* 2 *is free in* T.

Proof. Suppose on the contrary that there is a non-coloured edge which is not free. Clearly it does not belong to $\mathrm{pl}^\infty(T)$. So there is a maximum integer m such that $\mathrm{pl}^m T \neq \mathrm{pl}^\infty(T)$, $\mathrm{pl}^m(T)$ contains a non-coloured and non-free edge, and in $\mathrm{pl}^k(T)$ with $k > m$ each non-coloured edge is free.

Let e be any non-coloured edge from $E(\mathrm{pl}^m(T)) - E(\mathrm{pl}^{m+1}(T))$. Hence, according to Remark (5.4), e is incident with a hanging branching vertex, say v, of $\mathrm{pl}^m(T)$, and v has label 2. It suffices to show that e is free in $\mathrm{pl}^m(T)$. Consider two cases.

Case 1. The edge e_v of attachment of v either exists and is red or does not exist. Hence, by Lemma (6.1), if e_v exist then it is forbidden in $\mathrm{pl}^m(T)$. Therefore, by Step 2b of Procedure 2 and Remark (5.6), the edge e is the edge of attachment of one of three or more hanging strings attached to the vertex v. Now it is clear that e is free.

Case 2. There exists an edge e_v and e_v is non-coloured (possibly $e = e_v$). So, by Remark (5.6), e is one of three or more non-coloured edges incident to v. Let $e_v = vx$ where x is the second end-vertex of e_v and x belongs to $\mathrm{pl}^{m+1}(T)$. Since x is not hanging in T, the label of x is different from 1. Hence, by Remark (5.4), the label of x is either 12 or 21. Let $\mathrm{pl}^k(T)$ with $k \geqslant m+1$ be the tree with a hanging string containing x. If x has label 21, then in $\mathrm{pl}^k(T)$ there is exactly one green edge incident to x, since otherwise the edge e_v would be red. Hence, since $k > m$, by the assumption on m and by Lemma (6.2), in each case x is an end-vertex of a certain maximum path-factor in $\mathrm{pl}^k(T)$ and so in $\mathrm{pl}^{m+1}(T)$. Therefore, the edge $e_v = xv$ is not forbidden. On the other hand, owing to Corollary (3.11), it is not compulsory. Consequently, the edge e_v is free. Hence, one can see that also the edge e is free in $\mathrm{pl}^m(T)$. ■

Lemmas (6.1) and (6.3) imply

(6.4) THEOREM. *An edge of T is compulsory, forbidden, and free iff it is green, red and non-coloured, respectively.*

Now we shall prove the following

(6.5) LEMMA. *Any vertex v with label* 12 *is inner in a certain maximum path-factor of T.*

Proof. First consider the case where

$$V(\mathrm{pl}^\infty(T)) = \{v\}.$$

Then there is a minimal integer j such that $V(\mathrm{pl}^{j+1}(T)) = \{v\}$ and $j \geqslant 0$ since the label of v is not equal to 1. Since, moreover, v is labelled 12, in $\mathrm{pl}^j(T)$ there are at least 2 hanging branching vertices, say x and y, whose edges of attachment, say e_x and e_v, are incident to v. Let ${}^jS \in \mathscr{S}(F_1(\mathrm{pl}^j(T)))$. Then the vertex v is isolated in ${}^jS$ and according to formula (3.5), ${}^jS \in \mathscr{S}(\mathrm{pl}^j(T))$. On the other hand, each of the vertices x and y has label 2 and therefore, on the strength of Corollary (4.10), both x and y are inner in ${}^jS$. Let $e_1, e_2 \in E({}^jS)$ where e_1, e_2 are edges incident to x and to y, respectively. Then the vertex v is inner in the path-factor

$${}^jS - \{e_1, e_2\} \cup \{e_x, e_y\},$$

which belongs to $\mathscr{S}(\mathrm{pl}^j(T))$ since it has as many edges as ${}^jS$ has. Hence, by Corollary (3.12), the lemma follows.

Now consider the opposite case. According to Procedure 1, there is a minimal integer k, $k \geqslant 0$, such that the vertex v is incident to exactly one edge, say e, in $\mathrm{pl}^{k+1}(T)$. Therefore $e \in F_\infty(T)$ and, by Remark (5.2), the edge e is not red. Hence, by Theorem (6.4), e belongs to a certain ${}^{(k+1)}S \in \mathscr{S}(\mathrm{pl}^{k+1}(T))$. Owing to the definition of k, in $\mathrm{pl}^k(T)$ there is a hanging branching vertex, say x, which is adjacent to v and $\mathrm{pl}^{k+1}(T)$ is a component of $F_1(\mathrm{pl}^k(T))$. According to Corollary (3.2), there is a certain ${}^kS \in \mathscr{S}(F_1(\mathrm{pl}^k(T)))$ which contains ${}^{(k+1)}S$, v being of degree 1 in ${}^kS$. Exchanging only one edge in ${}^kS$, we can end the proof in a similar way as in the first case. ■

The following theorem gives the classification of vertices of a tree T with respect to their appearance in maximum path-factors of T.

(6.6) THEOREM. (i) *Each vertex with label* 1 *is hanging in all $S \in \mathscr{S}(T)$.*

(ii) *Each vertex with label* 2 *or* 3 *as well as each vertex with label* 21 *lying inside a green path is inner in all $S \in \mathscr{S}(T)$.*

(iii) *For each remaining vertex v (and so with label* 12 *or possibly* 21) *there are $S_1, S_2 \in \mathscr{S}(T)$ such that v is hanging in S_1 and inner in S_2.*

Proof. (i) is obvious, (ii) follows from Lemmas (4.9) and (6.1). To prove (iii) assume first that v is a vertex labelled 21 which is not inside a green path. Then, by Remark (5.7), v is incident to exactly one green edge of T. Consequently, from Lemmas (6.3) and (6.2) it follows that v is a hanging vertex in a certain maximum path-factor of T. Moreover, v is inner in another maximum path-factor of T since, according to Remarks (5.8) and (5.7) and to Theorem (6.4), v is incident to a compulsory edge as well as to another one, which is free in T. Thus (iii) holds true for that v. On the other hand, if v has label 12, then (iii) follows from Corollary (4.11) and Lemma (6.5).

Now it is easily seen that Theorem (6.6) makes it possible to determine the partition of vertices of the tree T into three classes consisting of vertices which in all maximum path-factors of T are, respectively,

1. always hanging,
2. always inner,
3. neither always hanging nor always inner.

Acknowledgment. The authors are indebted to the referees for suggesting improvements in the text of the paper.

References

[1] D. Barnette, *Trees in polyhedral graphs*, Canad. J. Math. 18 (1966), pp. 713–736.

[2] F. T. Boesch, S. Chen, and J. A. M. McHugh, *On covering the points of a graph with point disjoint paths*, in: Graphs and Combinatorics, Lecture Notes in Math. 406 (1974), pp. 201–212.

[3] S. Goodman and S. Hedetniemi, *On the Hamiltonian completion problem*, ibid. pp. 262-272.

[4] A. Kania, *Problem of determining all maximal path-factors of a tree by means of linear programming*, in: Graphs, Hypergraphs, and Block Systems, Proc. Symp. held in Zielona Góra, 1976 (Zielona Góra, 1976), pp. 113–118.

[5] L. Lovász, *On the structure of factorizable graphs*, Acta Math. Acad. Sci. Hungar. 23 (1972), pp. 179–195.

[6] — *The factorization of graphs*, II, ibid. 223–246.

[7] O. Ore, *Arc coverings of graphs*, Ann. Mat. Pura Appl. 55 (1961), pp. 315–321.

[8] Z. Skupień, *Path partitions of vertices and hamiltonity of graphs*, in: Recent Advances in Graph Theory, Proc. Symp. held in Prague, 1974 (Academia, Praha, 1975), pp. 481–491.

[9] — *Hamiltonian shortage, path partitions of vertices and matchings in a graph*, Colloq. Math. 36 (1976), pp. 305–318.

[10] — *Finding all maximal path factors of a tree and related problems*, in: Proc. Symp. held in Keszthely, 1976 (to appear).

[11] — and W. Zygmunt, *On generating all maximal path factors of a tree*, in: Graphs, Hypergraphs and Block Systems, Proc. Symp. held in Zielona Góra, 1976 (Zielona Góra, 1976), pp. 273–279.

[12] — — *Generating path factors of a tree*, Zeszyty Nauk. UJ 19 (1977), pp. 169–173.

[13] — — *Generating all maximal path-factors of a tree* (to appear).

INSTITUTE OF MATHEMATICS
ACADEMY OF MINING AND METALLURGY
Kraków

Accepté par la Rédaction le 13. 3. 1978