ZASTOSOWANIA MATEMATYKI
APPLICATIONES MATHEMATICAE
XIX, 3–4 (1987), pp. 599–610

M. SKOWROŃSKA (Toruń) and M. M. SYSŁO (Wrocław)

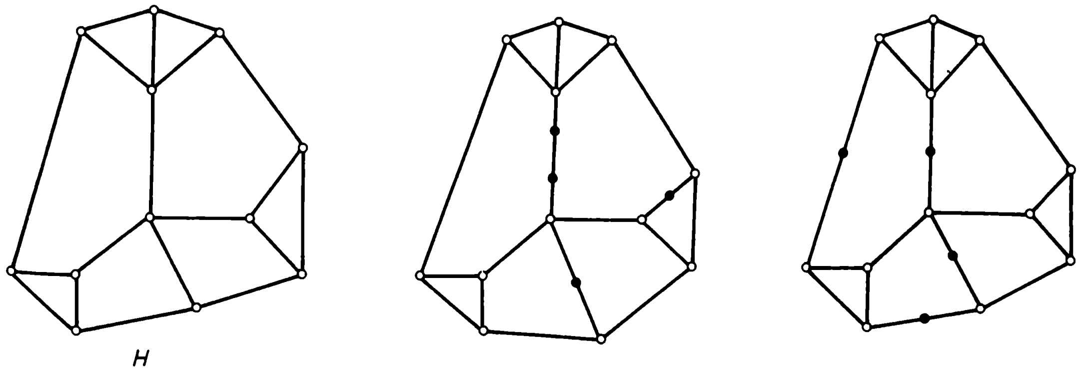
HAMILTONIAN CYCLES IN SKIRTED TREES

Abstract. A skirted tree can be obtained from a Halin graph by subdividing some of its interior edges. We characterize Hamiltonian skirted trees and those trees which can be interior trees of Hamiltonian skirted trees. Both characterizations are algorithmic and provide polynomial-time recognition algorithms for Hamiltonian skirted trees.

1. Introduction. A *Halin graph* H is a plane graph with

$$V(H) = V(T) \quad \text{and} \quad E(H) = E(T) \cup E(C),$$

where T is a plane tree with no vertices of degree 2, C is the cycle $(v_1, v_2, \ldots, v_k, v_1)$ and $v_1, v_2, \ldots, v_k$ are all leaves of T in a cyclic order. We write $H = T \cup C$ or $H(T)$ and call T an *interior tree* of H. If H is a Halin graph, then its interior plane tree is denoted by $T(H)$. Let $\mathscr{H}$ denote the family of all Halin graphs. Note that every wheel W is a Halin graph and in this case $T(W)$ is a star. Let $H \in \mathscr{H}$ and $T = T(H)$ have at least two nonleaves. We define two types of subgraphs in H. Let v be a nonleaf of T which is adjacent to only one other nonleaf of T. Let $C(v)$ denote the set of leaves of T adjacent to v, and $C'(v)$ denote the set $C(v)$ augmented with the two leaves of T adjacent along C to $C(v)$. A *fan* $F(v)$ with the centre v is the subgraph of H induced by $\{v\} \cup C(v)$, and an *extended fan* $F'(v)$ is the subgraph of H induced by $\{v\} \cup C'(v)$. It is evident that every Halin graph which is not a wheel contains at least two fans. Fig. 1 (a) shows a Halin graph with 3 fans.



(a) (b) (c)

Fig. 1

If $e = \{u, v\}$ is an edge in a graph G, then an *edge subdivision* of e results in replacing e in G by a series of edges. Let $\mathscr{F}$ denote the family of all graphs which can be obtained from Halin graphs by subdividing some of their edges. A graph G in $\mathscr{F}$ is called a *homeomorph* of a Halin graph, and $T(G)$ denotes the interior plane tree of G. For every $G \in \mathscr{F}$ there exists a unique Halin graph $H^c(G)$ which can be obtained from G by contracting each series of edges to an edge. A subgraph R of G is *a(n)* (*extended*) *fan* in G if the subgraph corresponding to R in $H^c(G)$ is a(n) (extended) fan.

A *skirted tree* is a homeomorph S of a Halin graph in which no exterior vertex has degree 2. Thus, S can be constructed from a plane tree T similarly as a Halin graph except that we allow T to have vertices of degree 2. We denote such a graph by $S(T)$. Let $\mathscr{S}$ denote the family of all skirted trees. Figs. 1 (b) and 1 (c) show two homeomorphs of the Halin graph H (Fig. 1 (a)), only the former of which is a skirted tree.

The purpose of this paper is to characterize Hamiltonian skirted trees. This goal is achieved in Section 2 where we characterize all Hamiltonian graphs in $\mathscr{F}$. In Section 3, we characterize those trees which are interior trees of Hamiltonian skirted trees. Both characterizations are algorithmic and provide polynomial-time recognition algorithms of Hamiltonian skirted trees.

2. Hamiltonian skirted trees. It is easy to show that every Halin graph is Hamiltonian (see [1]). To this end, let $F(v)$ be a fan in a Halin graph H different from a wheel and let H_v denote the graph obtained from H by shrinking $F(v)$ to one vertex. That is, the vertices of H_v are all vertices of H not in $F(v)$, together with the new vertex $\bar{v}$ corresponding to $F(v)$; the edges are all edges of H which do not belong to $F(v)$; the vertices incident with a given edge are the same as in G, unless one was in $F(v)$ in which case that end of the edge becomes now $\bar{v}$. Fig. 2 illustrates this operation. It is easily seen that H_v is also a Halin graph for every fan $F(v)$ of H.

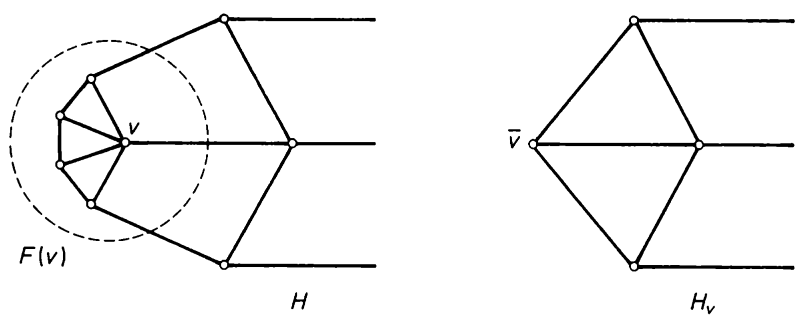


Fig. 2

Moreover, this reduction of a Halin graph preserves the existence of a Hamiltonian cycle. That is, every Hamiltonian cycle in H generates a Hamiltonian cycle in H_v and, conversely, a Hamiltonian cycle of H_v can be extended to a Hamiltonian cycle in H. Therefore, we can again apply the reduction process to H_v and to one of its fans. Continuing, we finally reach a wheel. Since every wheel is Hamiltonian, every Halin graph is Hamiltonian.

Our aim now is to define a similar reduction procedure for skirted trees. Let G be a homeomorph of a Halin graph H. We allow also exterior edges of H to be subdivided in G, since such graphs G may occur in the process of verifying whether a skirted tree is Hamiltonian. For the sake of simplicity, no two vertices of degree 2 are adjacent in G. If G or its reduced copy contains a series p of more than two edges, then, without loss of generality, we may contract p to a series of length 2.

We first determine, in terms of forbidden patterns of subdivided edges, which homeomorphs of wheels are Hamiltonian.

LEMMA 1. *A homeomorph G of a wheel is Hamiltonian if and only if G contains:*

(i) *no two consecutive interior edges which are subdivided,*
(ii) *no three subdivided edges incident with a vertex,*
(iii) *no three subdivided edges which are mutually adjacent.*

Proof. The lemma follows from the form of a Hamiltonian cycle h in a wheel: h contains all exterior edges except one and two interior edges which are adjacent to the exterior edge which is missed in h.

It is easy to see that (i)–(iii) are also forbidden patterns of subdivided edges for an extended fan in $G \in \mathscr{F}$ to be reducible with a preserved Hamiltonian cycle. First note that if an extended fan $F'(v)$ in G contains one of the patterns (i)–(iii), then G is not Hamiltonian. On the other hand, if G has an extended fan $F'(v)$ which contains no patterns (i)–(iii), then G, by shrinking the fan $F(v)$, can be reduced to a graph G_v such that $G_v \in \mathscr{F}$ and, moreover, G is Hamiltonian if and only if G_v is Hamiltonian. Basic reductions of $F(v)$ are shown in Fig. 3 (a)–(e). For instance, the reduction illustrated in Fig. 3 (b) means that if the exterior rib of $F(v)$ is subdivided and G is Hamiltonian, then every Hamiltonian cycle in G must contain edge $\{w', w\}$. Hence, if G is reduced to G_v by shrinking $F(v)$ and every Hamiltonian cycle in G_v is to generate a Hamiltonian cycle in G, then every Hamiltonian cycle in G_v must contain edge $\{\bar{v}, w\}$. This requirement for G_v is met by subdividing $\{\bar{v}, w\}$. The reduction of fans whose extensions contain other feasible patterns of subdivided edges can be defined by combining the reduction rules defined in Fig. 3 (a)–(e). For instance, see Fig. 4.

We warn the reader that before a fan $F(v)$ of G is reduced, it is necessary to test that its extended fan $F'(v)$ contains no pattern (i)–(iii) of subdivided edges. Otherwise, the extended fan of Fig. 4 with the edge $\{x_1, x_2\}$ subdivided additionally could be reduced by applying the same steps.

We conclude this section with the theorem which summarizes the discussion above.

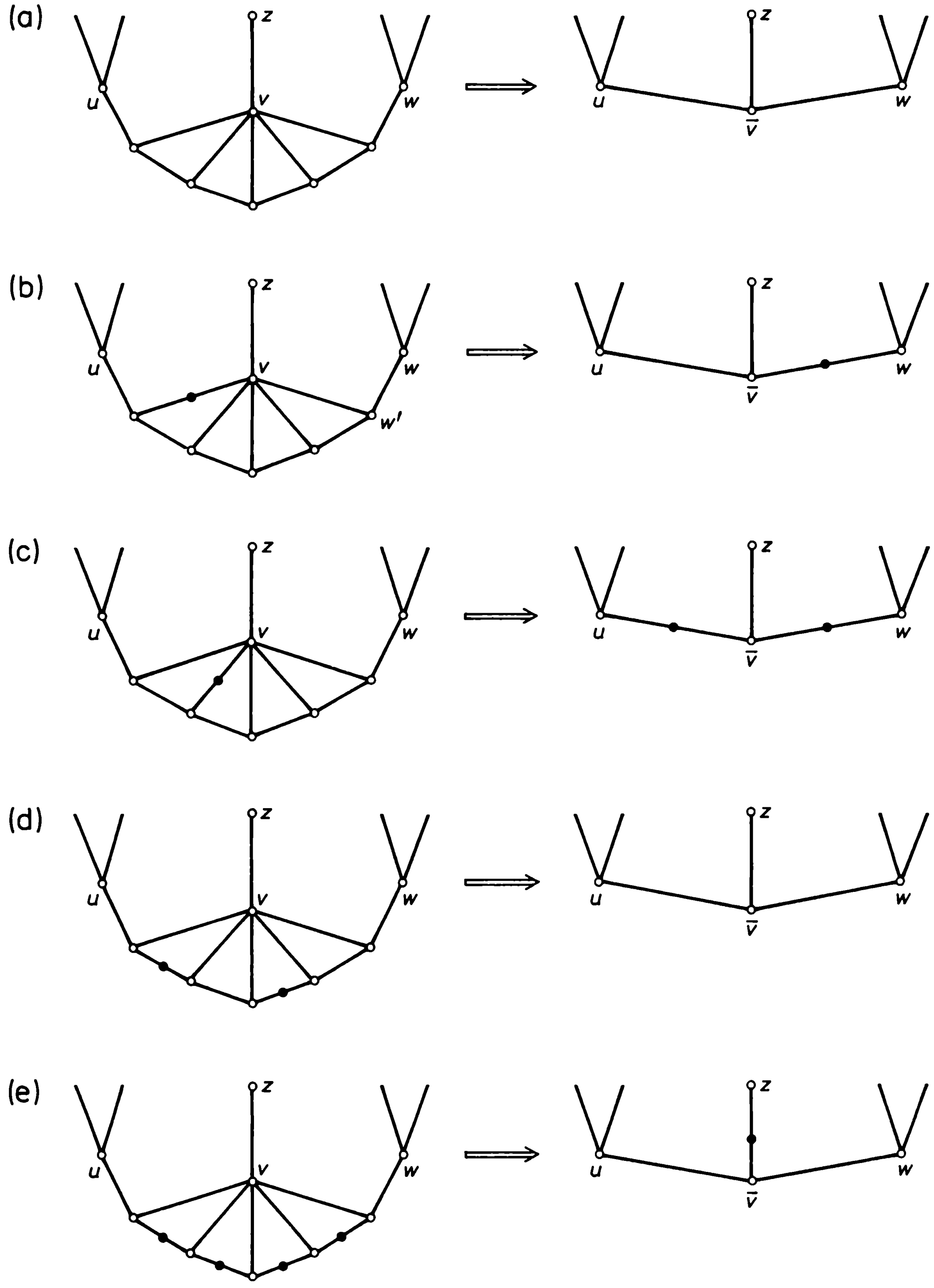


Fig. 3

THEOREM 1. *Let G be a homeomorph of a Halin graph and $F'(v)$ be an extended fan in G. Then G is Hamiltonian if and only if $F'(v)$ contains no patterns of subdivided edges* (i)–(iii) (*listed in Lemma* 1) *and the reduced graph G_v obtained from G by applying the rules of Fig.* 3 (a)–(e) *is Hamiltonian.*

Proof. It is evident that if an extended fan $F'(v)$ of G contains one of the patterns (i)–(iii), then G is not Hamiltonian. Otherwise, every pattern of

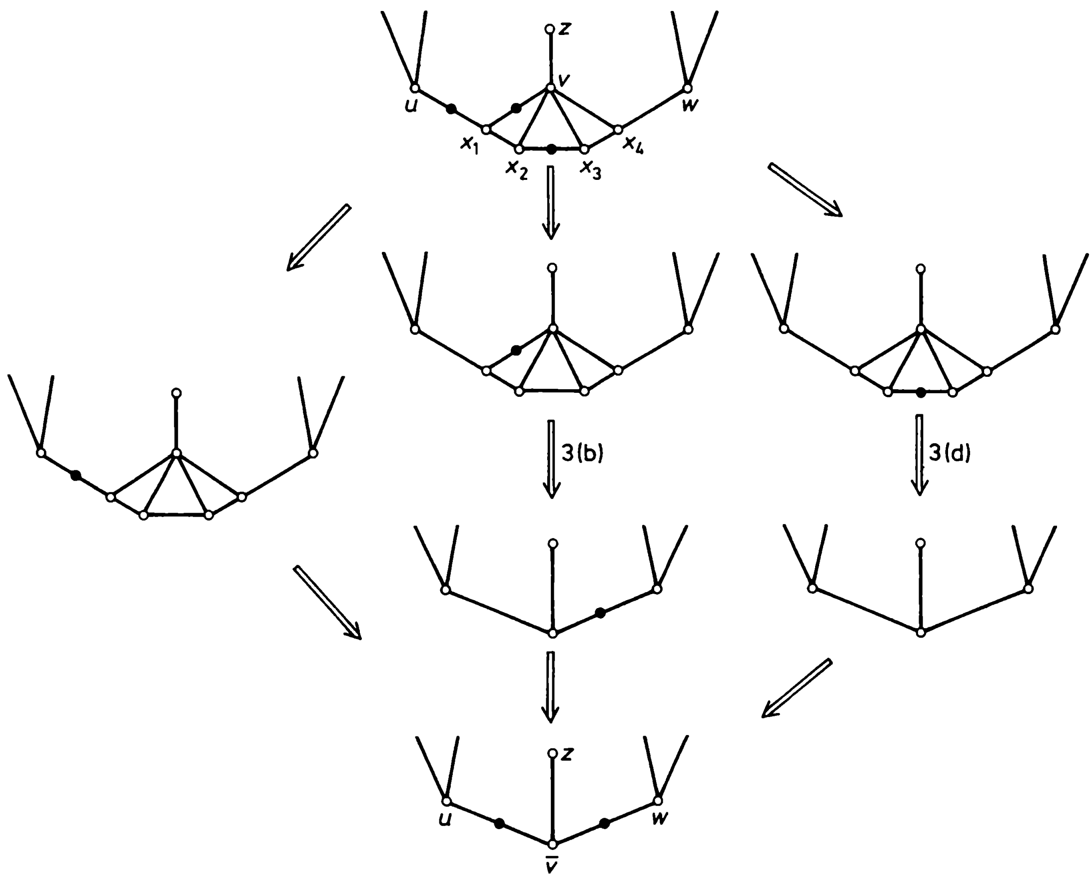


Fig. 4

subdivided edges in $F(v)$ can be partitioned into a number of elementary cases listed in Fig. 3 (a)–(e), where it is shown also how these elementary patterns can be reduced preserving the property that a graph is Hamiltonian. If a particular feasible pattern of subdivided edges in $F(v)$ has to be partitioned into elementary ones, the results of elementary reductions have to be combined into one pattern in the reduced graph G_v. We illustrate such a process in Fig. 4. (Note that all subdivided edges in $F'(v)-F(v)$ must be carried over to G_v.) The graph G_v is also a homeomorph of a Halin graph. By the definition of reductions, each Hamiltonian cycle in G generates a Hamiltonian cycle in G_v and every Hamiltonian cycle in G_v can be extended to that in G. Hence, we reach the theorem conclusion.

Theorem 1 justifies one reduction step of a graph G in $\mathscr{F}$. The reduction can be continued until either a current graph contains one of the patterns (i)–(iii) or it has only one fan. In the former case G is not Hamiltonian, and in the latter case G is Hamiltonian provided the resulting wheel has no pattern (i)–(iii).

In the next section we characterize trees which are interior trees of Hamiltonian skirted trees.

3. Trees which can generate Hamiltonian skirted trees. There exist trees which have no embeddings generating Hamiltonian skirted trees (see Fig. 5 (a)). Fig. 5 (b) shows however a plane tree T' such that $S(T'')$ is not Hamiltonian but T' has an embedding T'' (Fig. 5 (c)) for which $S(T')$ is Hamiltonian. The aim of this section is to characterize those trees which can be embedded so that the resulting skirted trees are Hamiltonian. (Note that such trees have necessarily at least one vertex of degree greater than 2.) To this end, we shall utilize the correspondence between Hamiltonian cycles in $G \in \mathcal{S}$ and certain path partitions of $T(G)$. Let $\mathcal{C} = \{C_i\}$ be the set of interior faces (considered as cycles) in G and let C_i' denote the path $C_i - e_i$, where e_i is the exterior edge of C_i. It is easy to see (we proved this correspondence for Halin graphs in [7]) that there exists a one-to-one correspondence between Hamiltonian cycles in G and path partitions $\mathcal{P}$ of the vertices of $T(G)$, where

$$\mathcal{P} \subset \{C_i'\} \cup \{(v_j)\colon\ v_j \text{ is a leaf in } T(G)\}.$$

In other words, a Hamiltonian cycle in G partitions the vertex set $V(G) = V(T(G))$ into vertex-disjoint paths whose both ends are consecutive leaves in $T(G)$. We assume in what follows that each leaf v forms a path (v) of length 0.

If T is a tree and $\mathcal{P}$ is any partition of $V(T)$ into vertex-disjoint paths whose both ends are leaves in T (we call such a path partition of T *feasible*), then there exists an embedding $\tilde{T}$ of T in the plane such that the end vertices of paths in $\mathcal{P}$ are consecutive in $\tilde{T}$. To show this, let T' be an embedding of T and $p_i \in \mathcal{P}$ be a path whose end vertices are not consecutive. We now form an embedding T'' of T in which the end vertices of p_i are consecutive and the relative positions of end vertices of the other paths are not altered. Thus, we can reach an embedding $\tilde{T}$ in which the end vertices of all paths $p_i \in \mathcal{P}$ are consecutive.

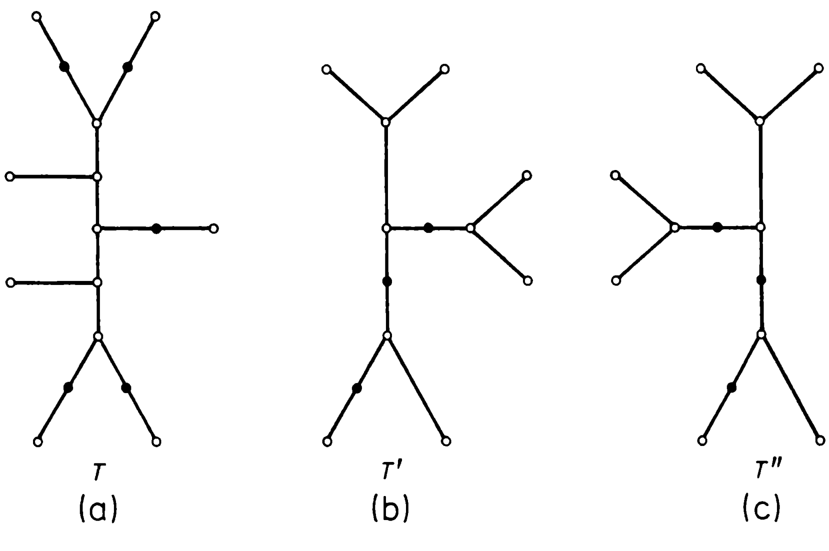


Fig. 5

The problem now is to find which trees T admit feasible path partitions $\mathscr{P}$. Every edge incident with a vertex of degree 2 belongs to some path in $\mathscr{P}$, hence no vertex is incident with more than two such edges. If $v \in V(T)$ is incident with exactly two such edges, then each other edge incident with v belongs to no path in $\mathscr{P}$ and can be removed from T. The removal of some edges from T may result in new vertices of degree 2 and in some vertices of degree at most 1. In the former case the reduction process can be continued and the latter T has no feasible path partition if a vertex of degree at most 1 in the reduced graph was not a leaf in the original tree T. We shall prove that if the reduction process does not produce new vertices of degree at most 1, then T has a feasible path partition. To be more precise, let us first formalize the reduction process in the form of an algorithm in a Pascal-like notation.

ALGORITHM REDUCTION. {Given a tree T different from a path. The algorithm returns *fail* = *true* if T has no feasible path partition. Otherwise, it returns a subgraph T_R of T which contains a feasible path partition. If *success* = *true*, then T_R is the feasible path partition of T.}

```
    begin
      success ← false; fail ← false; T_R ← T;
      for each vertex v in T_R do
        if v is a leaf then l(v) ← p else l(v) ← n;
      repeat
        for each edge e in T_R incident with a vertex of degree 2 do l(e) ← r;
          if T_R has a vertex incident with more than two r-edges
1.          then fail ← true
            else begin
2.              T_R ← T_R − {e = (x_1, x_2) ∈ E(T_R): e is not an r-edge and x_1
                                                  or x_2 is incident with exactly
                                                  two r-edges.};
                if T_R has a vertex v of degree at most 1 and l(v) = n
3.                 then fail ← true
                   else if each component of T_R is a path
                          then success ← true
                end
      until success or fail or (all edges incident with vertices of degree 2 are
                                labelled)
    end
```

PROPERTY 1. *Every edge of a tree T labelled r by the algorithm REDUCTION belongs to every feasible path partition of T.*

Proof. An edge e of T is labelled r by the algorithm if e is incident with a vertex of degree 2 either in the original tree T or in its subtree obtained by the removal from T of those edges which do not belong to any feasible path partition (statement 2). Hence such an edge e must be included in every feasible path partition.

PROPERTY 2. *If the algorithm REDUCTION applied to a tree T terminates with fail = true, then T has no feasible path partition.*

Proof. The algorithm assings $fail \leftarrow true$ either when T has a vertex v incident with at least three edges which must belong to every feasible path partition of T (statement 1) or when a nonleaf v of T becomes of degree at most 1 (statement 3). In both cases, T has no feasible path partition which contains v.

We now show that if the algorithm terminates with $fail = false$, then T_R has a feasible path partition, and hence also T has. To this end, we will utilize the following properties of T_R which follow directly from the algorithm description.

PROPERTIES 3. *If the algorithm REDUCTION applied to a tree T terminates with fail = false, then T_R has the following properties* ($\deg_R(v)$ *denotes the degree of v in T_R*):

(i) *Each component of T_R is a tree or a path and $V(T_R) = V(T)$.*

(ii) *For each vertex v in T_R:*

if $\deg_R(v) \leqslant 1$, *then* $l(v) = p$;

if $\deg_R(v) = 2$, *then v is incident with two r-edges*;

if $\deg_R(v) \geqslant 3$, *then v is incident with at most one r-edge.*

(iii) *Each r-edge of T_R is incident with at least one vertex of degree* 2.

LEMMA 2. *Every graph G which has Properties* 3 (i)–(iii) *contains a feasible path partition.*

Proof. It is sufficient to prove this lemma for each connected component of G which is not a path. Let F be a component of G. We proceed by induction on m, the number of vertices of degree greater than 2 in F.

Let $m = 1$ and $\deg_F(v) = d \geqslant 3$ for $v \in V(F)$. Then F is either a star or a star with a tail (see Fig. 6 (b)). Evidently, such an F has a feasible path partition.

Let $m = k$, $k \geqslant 2$, in F and assume that every graph F' with $m(F') < k$ has a feasible path partition. F contains a vertex v such that $\deg_F(v) \geqslant 3$ and v is adjacent to exactly one vertex of degree at least 2. Let u denote a vertex of F satisfying the following condition:

$$d_F(v, u) = \min\{d_F(v, w)\colon\ w \in V(F) \text{ and } \deg_F(w) \geqslant 3\}.$$

Since F is a tree, with $m \geqslant 2$, such vertices u and v always exist and F is either as in Fig. 6 (c) or as in Fig. 6 (d).

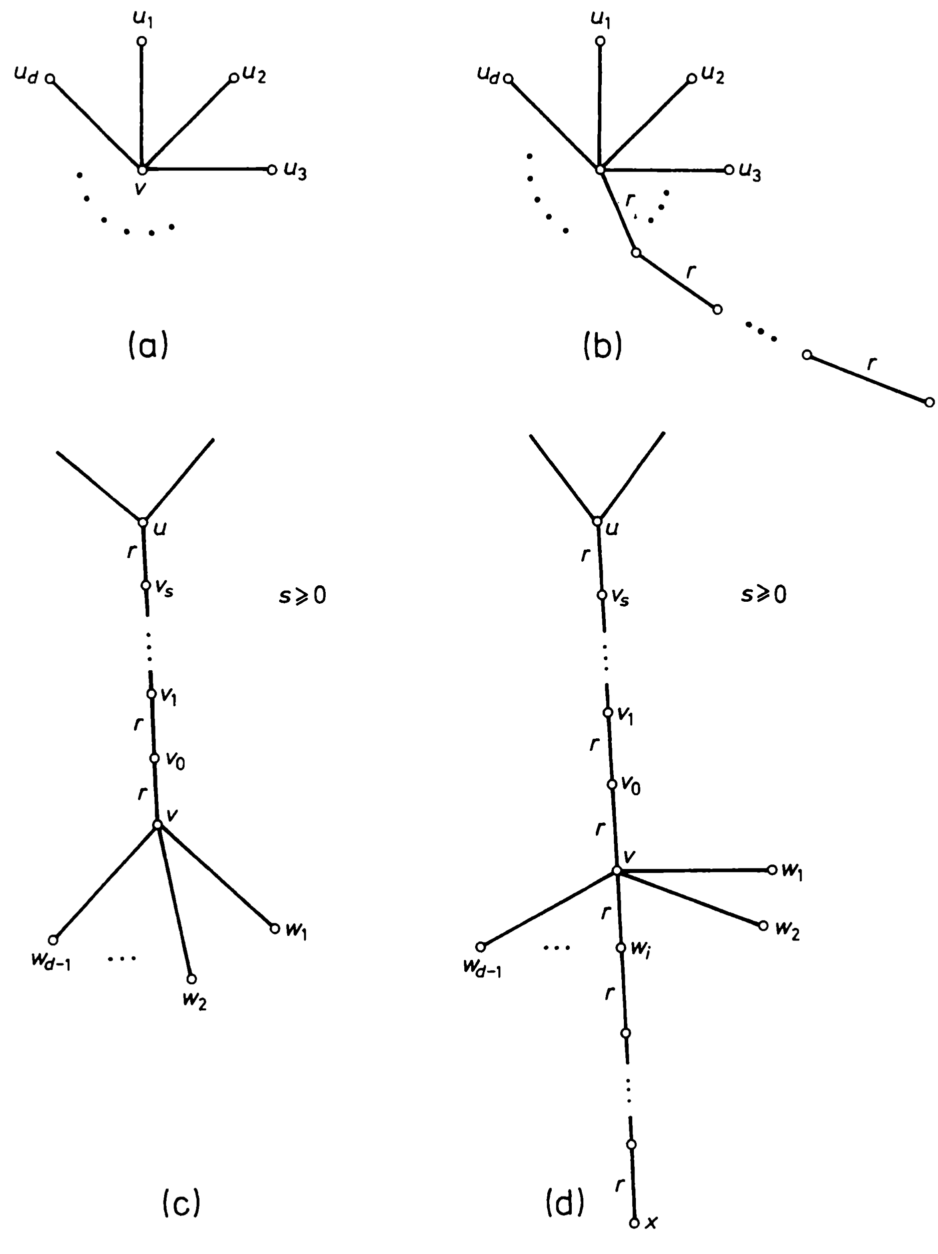


Fig. 6

In the former case we put

$$F' = F - \{w_1, w_2, \ldots, w_{d-1}\}.$$

F' has $m-1$ vertices of degree at least 3 and has Properties 3 (i)–(iii). By the inductive assumption, F' contains a feasible path partition $\mathscr{P}'$. If v forms a path $\mathscr{P}'$, then

$$\mathscr{P} = \mathscr{P}' - \{(v)\} \cup \{(w_1, v, w_2)\} \cup \bigcup_{j=3}^{d-1} \{(w_j)\}$$

is a feasible path partition of F. If v is an end vertex of a path p in $\mathscr{P}'$, then to form $\mathscr{P}$ we extend p to (p, w_1) and add the paths (w_j) for $j = 2, \ldots, d-1$.

In the latter case, we put

$$F' = F - (W \cup \{w_1, \ldots, w_{i-1}, w_{i+1}, \ldots, w_{d-1}\}),$$

where W is the set of vertices of the path from x to w_i. Similarly as above, F' satisfies the inductive assumptions, so it has a feasible path partition which can be easily extended to that of F.

We can now summarize:

THEOREM 2. *A tree T different from a path has a feasible path partition if and only if the algorithm REDUCTION applied to T terminates with fail $=$ false.*

Proof. By Property 2, if the algorithm terminates with *fail* $=$ *true*, then T has no feasible path partition. Conversely, if *fail* $=$ *false*, then by Properties 3 (i)–(iii) and Lemma 2, T_R has a feasible path partition which is also a feasible path partition of T.

We now present an algorithm which generates a feasible path partition in T when the algorithm REDUCTION returns *fail* $=$ *false*.

ALGORITHM CONSTRUCTION. {Assume that T_R is the output of the algorithm REDUCTION which applied to a tree T returns *fail* $=$ *false*. This algorithm reduces further T_R to a feasible path partition of T.}

```
   begin
     while T_R is not a collection of paths do
     begin
       x ← a leaf in a component of T_R which is not a path;
1.     y ← a vertex of degree greater than 2 closest to x;
       f ← the last edge on the path from x to y;
2.     if f = {x, y} then l(f) ← r;
       let e_1, e_2, ..., e_k (k ≥ 2) be the edges incident with y and e_i ≠ f;
3.     if no e_i (i = 1, 2, ..., k) is an r-edge then l(e_1) ← r;
       while T_R has a vertex incident with two r-edges do
       begin
4.       T_R ← T_R − {e = {x_1, x_2} ∈ E(T_R): e is not an r-edge and x_1 or x_2 is
                                              incident with exactly two r-edges};
5.       for each edge e in T_R incident with a vertex of degree 2 do l(e) ← r
       end
     end
   end
```

The behaviour of the algorithm CONSTRUCTION is summarized in the following theorem:

THEOREM 3. *Let T be a tree different from a path. If the algorithm REDUCTION applied to T returns fails = false and the subgraph T_R of T, then the algorithm CONSTRUCTION produces a feasible path partition of T.*

Proof. Let $\mathscr{P}$ denote the subgraph T_R of T produced by the algorithm CONSTRUCTION. It is clear that $\mathscr{P}$ is a collection of paths of the tree T (we remind the reader that a single vertex is also a path). To complete the proof it remains to show that both end vertices of each path in $\mathscr{P}$ are leaves in T. Let us assume that $\mathscr{P}$ contains a path p with at least one end vertex which is a nonleaf in T. Let us put

$$p = (u, v_1, v_2, \ldots, v_k, w)$$

and let $l(u) = n$. Since in the input T_R to the algorithm (equivalently, in T_R produced by the algorithm REDUCTION) every vertex v of degree at most 1 is a leaf of T, we have $\deg_R(u) \geqslant 2$.

If $\deg_R(u) = 2$, then both edges incident with u are r-edges in T_R and none of them is removed from T_R by the algorithm CONSTRUCTION, a contradiction.

Let $\deg_R(u) \geqslant 3$. Note first that the algorithm labels with r at most two edges incident with each vertex (statements 2, 3, and 5), hence no r-edge is removed from T_R (statement 4). If u is a vertex y for some vertex x (statement 1), then it becomes of degree 2 and both edges incident with u, which remain, are r-edges, so they are not removed in next steps. If u is adjacent to such a vertex y, then its degree is decremented by 1. Therefore, if finally u becomes of degree 2, then both edges incident with u, which remain, become r-edges (statement 5), so they are not removed in the next steps. Thus, we also arrive at a contradiction.

4. Conclusions. We were able to characterize Hamiltonian skirted trees, which generalize Halin graphs, and to characterize those trees which have an embedding in the plane so that the resulting skirted trees are Hamiltonian. Both characterizations are algorithmic in a sense that we presented efficient (i.e., working in polynomial time) methods for recognizing these graphs. It remains however to settle if there exist structured characterizations, e.g., in terms of forbidden subgraphs.

It is easy to see that some of the results proved for Halin graphs can be extended to Hamiltonian skirted trees. For instance, the travelling salesman problem (TSP) has a polynomial-time algorithm for Halin graphs (see [2]) and a similar approach can be used to solve the TSP on skirted trees. A special version of the set-partitioning problem resulting from the Hamiltonian cycle problem on Halin graphs can be also easily generalized to that which corresponds to the Hamiltonian cycle problem on skirted trees; see [3] and [7] for details. All Halin graphs are almost pancyclic (see [5]), it is however unlikely that also many skirted trees share this property. The graphs which can be obtained from Halin graphs by contracting some of the exterior edges are also Hamiltonian (see [4]). It would be interesting to investigate which exterior contractions of skirted trees are Hamiltonian.

The vertex- and edge-coloring problems on skirted trees have linear-time solution method (see [6]).

References

[1] J. A. Bondy, *Pancyclic graphs: recent results*, pp. 180–187 in: A. Hajnal, R. Rado and V. T. Sós (eds.), *Infinite and Finite Sets* (Colloq. Math. Soc. János Bolyai, Vol. 10), Kiado, Budapest 1975.

[2] G. Corneújols, D. Naddef and W. R. Pulleyblank, *Halin graphs and the travelling salesman problem*, Math. Programming 26 (1983), pp. 287–294.

[3] J. Malkevitch, *Cycle lengths in polytopal graphs*, pp. 364–370 in: Y. Alavi and D. R. Lick (eds.), *Theory and Applications of Graphs*, Springer-Verlag, Berlin–Heidelberg 1978.

[4] M. Skowrońska, *Hamiltonian properties of Halin-like graphs*, Ars Combinatoria 16-B (1983), pp. 97–109.

[5] – *The pancyclicity of Halin graphs and their exterior contractions*, in: B. Alspach and C. Godsil (eds.), *Cycles in Graphs*, Ann. Discrete Math. 27 (1985), pp. 179–194.

[6] – *Efficient vertex- and edge-coloring of Halin graphs*, pp. 273–277 in: M. Fiedler (ed.), *Graphs and Other Combinatorial Topics*, Teubner-Texte zur Math., Band 59, Leipzig 1983.

[7] M. M. Sysło, *On two problems related to the traveling salesman problem on Halin graphs*, pp. 325–335 in: G. Hammer and D. Pallaschke (eds.), *Selected Topics in Operations Research and Mathematical Economics*, Springer-Verlag, Berlin–Heidelberg 1984.

INSTITUTE OF MATHEMATICS
NICOLAUS COPERNICUS UNIVERSITY
87-100 TORUŃ

INSTITUTE OF COMPUTER SCIENCE
UNIVERSITY OF WROCŁAW
51-151 WROCŁAW