ADVANCES IN APPLIED MATHEMATICS **18,** 33–49 (1997)
ARTICLE NO. AM960495

Interval Partitions and Activities for the Greedoid Tutte Polynomial

Gary Gordon and Elizabeth McMahon

Lafayette College, Easton, Pennsylvania 18042

The two variable greedoid Tutte polynomial $f(G; t, z)$, which was introduced in previous work of the authors, is studied via external activities. Two different partitions of the Boolean lattice of subsets are derived and a feasible set expansion of $f(G)$ is developed. All three of these results generalize theorems for matroids. One interval partition yields a characterization of antimatroids among the class of all greedoids. As an application, we prove that when G is the directed branching greedoid associated with a rooted digraph D, then the highest power of $(z + 1)$ which divides $f(G)$ equals the minimum number of edges which can be removed from D to produce an acyclic digraph in which all vertices of D are still accessible from the root. The unifying theme behind these results is the idea of a computation tree.

1. INTRODUCTION

The Tutte polynomial is an extremely interesting and well studied invariant from graphs and matroids. Many classical counting problems can be expressed in terms of this polynomial, from the chromatic number of a graph to the number of regions into which a family of hyperplanes divides Euclidean space. An extensive introduction to the theory surrounding this invariant can be found in [4].

There are three equivalent ways to define the Tutte polynomial of a matroid: in terms of a deletion-contraction recursion, in terms of a subset expansion (the corank-nullity generating function) and via basis activities (which was Tutte's original approach). In generalizing the Tutte polynomial from matroids to greedoids in [8], we have taken the subset expansion viewpoint (Eq. (1.2) below) and found a deletion-contraction recursion (Eq. (1.3) below). The two-variable polynomial $f(G; t, z)$ (or simply $f(G)$) which results also has the multiplicative direct sum property: $f(G_1 \oplus G_2) = f(G_1)f(G_2)$. This polynomial has been studied for several greedoid classes in [5–8, 12].

0196-8858/97 $25.00

In [2], Björner, Korte, and Lovász use the activities approach to define a one-variable greedoid polynomial $\lambda(G)$. This approach, for matroids, has close connections with combinatorial topology, especially with the shelling of various simplicial complexes associated with the matroid. Since $\lambda(G)$ is an evaluation of the two variable polynomial $f(G)$, this leads to the question of how $f(G)$ can be viewed in terms of activities in a greedoid. This question is central to this paper.

In Section 2, we define a computation tree for a greedoid based on the recursive process of computing $f(G)$ using deletion and contraction. This idea is certainly not new; any attempt to compute the Tutte polynomial recursively will give rise to one. Despite this, we feel that the tree itself is important; it provides the basis for most of the results in this paper. For example, these trees are used in [9] when G is a matroid to unify several different formulations of the Tutte polynomial. The main results in this section, Theorem 2.5 and Proposition 2.7 concern two distinct interval partitions of the Boolean lattice which have direct applications to the polynomial $f(G)$. Both of these results can be viewed as generalizations of a theorem of Crapo for matroids. We also show how a computation tree can be used to give an algorithmic characterization of antimatroids (Proposition 2.4).

In Section 3, we apply 2.5 to $f(G)$ to obtain a new feasible set expansion for $f(G)$ (Theorem 3.1). We also make explicit the connection between basis activities of [2] and the computation tree.

Section 4 is concerned with a direct application of the computation tree idea to directed branching greedoids. We write $G(D)$ for the directed branching greedoid associated with a rooted digraph D. The main result (Theorem 4.2) shows that if $(z+1)^k$ divides $f(G(D); t, z)$ with k maximum, then k is the minimum number of edges which can be removed from D to create a spanning acyclic rooted digraph.

We now review some of the mathematical preliminaries we will need. We will assume the reader is familiar with matroids; much of the greedoid terminology is borrowed directly from matroid theory. See Björner and Ziegler [3] or Korte, Lovász, and Schrader [2] for extensive background material on greedoids.

DEFINITION 1.1. A *greedoid* G is a pair $(E, \mathscr{F})$, where E is a finite set and $\mathscr{F}$ is a family of subsets of E satisfying:

$G1$. For every nonempty $X \in \mathscr{F}$, there is an element $x \in X$ such that $X - \{x\} \in \mathscr{F}$;

$G2$. For $X, Y \in \mathscr{F}$ with $|X| < |Y|$, there is an element $y \in Y - X$ such that $X \cup \{y\} \in \mathscr{F}$.

A subset $X \in \mathscr{F}$ is called *feasible*. The *bases* of G are the maximal feasible sets; property $G2$ ensures that the bases are equicardinal. The *rank* of a subset $S \subseteq E$, written $r(S)$, is the size of the largest feasible subset of S. A subset $S \subseteq E$ is *spanning* if S contains a basis. A *loop* in a greedoid is an element which is in no feasible set. We occasionally will call these greedoid loops, especially when discussing digraphs, to distinguish them from graph theoretic loops. A *coloop* is an element which is in every basis of G. Graph theoretic interpretations of these concepts are explored in Section 4.

If G is a greedoid on the ground set E, with feasible sets $\mathscr{F}$ and rank function r, then we define the two-variable greedoid polynomial $f(G; t, z)$ by

$$f(G; t, z) = \sum_{S \subseteq E} t^{r(G) - r(S)} z^{|S| - r(s)}. \tag{1.2}$$

This is the corank-nullity formulation of the Tutte polynomial when G is a matroid. $f(G)$ satisfies the following fundamental deletion-contraction recursion; this underlies all of the recursive structures used throughout this paper.

$$\text{If } \{e\} \in \mathscr{F}, \quad \text{then } f(G; t, z) = f(G/e) + t^{r(G) - r(G-e)} f(G - e). \tag{1.3}$$

We will need the following characterization of antimatroids. These characterizations appear in [2, (Lemmas 1.2 and 1.7)].

PROPOSITION 1.4. *Let $G = (E, \mathscr{F})$ be an accessible set system. Then the following are equivalent*:

(*a*) *G is an antimatroid,*

(*b*) *If $F, F \cup \{x\}$ and $F \cup \{y\} \in \mathscr{F}$, then $F \cup \{x, y\} \in \mathscr{F}$,*

(*c*) *If $F_1, F_2 \in \mathscr{F}$ with $F_1 \subseteq F_2$, then $F_1 \cup \{x\} \in \mathscr{F}$ implies $F_2 \cup \{x\} \in \mathscr{F}$.*

2. COMPUTATION TREES AND INTERVAL PARTITIONS

If G is a greedoid, a computation tree $T(G)$ will be a rooted binary tree in which each vertex is labeled by a minor of G in the following way: the two children of a vertex receiving label G' will be labeled G'/e and $G' - e$, where $\{e\}$ is feasible in G'.

DEFINITION 2.1. A *computation tree* $T(G)$ for a greedoid G is a rooted labeled tree defined as follows:

(a) If G has no feasible elements (i.e., each element of G is a loop), then $T(G)$ is the trivial labeled rooted tree with a single vertex, labeled by G.

(b) If $\{e\}$ is feasible in G, then recursively obtain $T(G)$ by forming the rooted labeled trees for G/e and $G - e$ and joining them as in Fig. 1.

By applying (b) recursively to each leaf of the tree labeled by a greedoid of positive rank, this process will always terminate with the leaves of the computation tree receiving labels which correspond to rank 0 greedoid minors of G (i.e., each leaf of $T(G)$ will correspond to a collection of loops).

EXAMPLE 2.2. Let G be the directed branching greedoid associated with the rooted digraph at the top of Fig. 2. Figure 2 then gives a computation tree for G. (Directed branching greedoid deletion and contraction correspond to the usual definition of deletion and contraction from graph theory.) We use the convention that the left-child of a vertex is obtained by contraction and the right child comes from deletion.

For a computation tree $T(G)$ with m leaves we introduce the following notation. Let $\{G_k: 1 \leq k \leq m\}$ be the set of (rank 0) greedoid minors of G which label the leaves of $T(G)$; for each k, let F_k represent the elements of G which were contracted in the unique path from the root (labeled by G) to the leaf labeled by G_k, and let $\text{ext}_T(F_k)$ represent the elements of G which correspond to loops in G_k. (We order the labeled leaves from left to right, using the "contraction precedes deletion" convention of Fig. 2.) Thus, $\text{ext}_T(F_k)$ is the set of elements of G which were neither deleted nor contracted along the path in $T(G)$ from the root to the leaf corresponding to F_k. We will call $x \in \text{ext}_T(F_k)$ *externally active* for F_k with respect to $T(G)$. For example, $\text{ext}_T(\{a, c\}) = \{b\}$, $\text{ext}_T(\phi) = \{c\}$ and $\text{ext}_T(\{a\}) = \phi$ for $T(G)$ in Fig. 2. Different computation trees can give rise to different sets $\text{ext}_T(F_k)$; for example, if $((G - b)/a) - c$ labels a vertex in a different computation tree $T'(G)$, then $\text{ext}_{T'}(\{a, c\}) = \phi$ (although $\text{ext}_T(\phi) = \{c\}$ for any tree T). Finally, recall that the closure of $S \subseteq E$, denoted by $\sigma(S)$, is defined by $\sigma(S) := \{x \in E: r(S \cup \{x\}) = r(S)\}$.

PROPOSITION 2.3. *Let $T(G)$ be any computation tree for the greedoid G and let F_k and $\text{ext}_T(F_k)$ be defined as above. Then $\{F_k: 1 \leq k \leq m\}$ is precisely the collection of all feasible sets of G and $\text{ext}_T(F_k) \subseteq \sigma(F_k) - F_k$.*

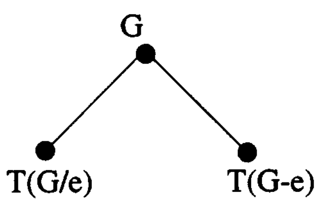

FIGURE 1

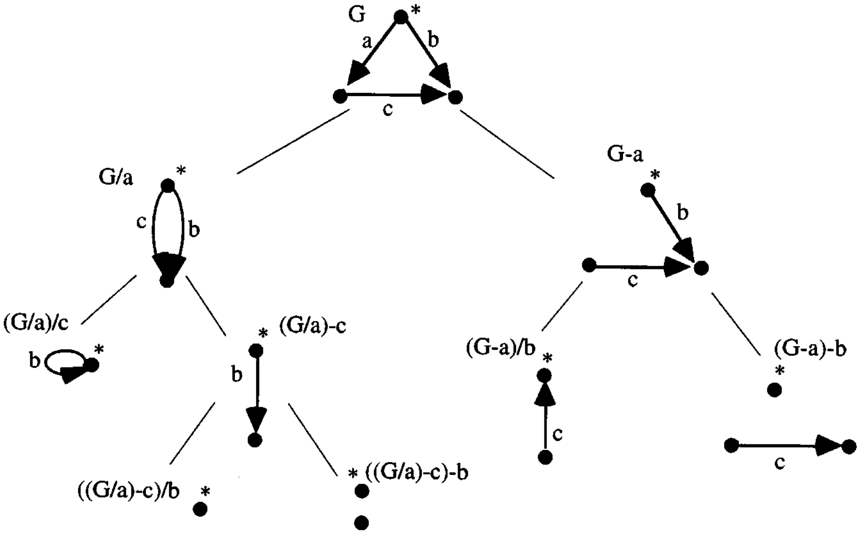

FIGURE 2

Proof. If G has no feasible singletons, then every element of G is a loop. Then $T(G)$ is the trivial, single-vertex tree, ϕ is the only feasible set and the theorem holds.

We may now assume G contains a feasible set $\{e\}$ and use induction on $|E| = n$. When $n = 1$, $T(G)$ has two leaves, which are both labeled by the empty greedoid, one corresponding to G/e and the other corresponding to $G - e$. In this case, we have $F_1 = \{e\}$, $\text{ext}_T(F_1) = \phi$, $F_2 = \phi$, and $\text{ext}_T(F_2) = \phi$. This tree is unique and clearly satisfies the theorem. Now assume that $n > 1$ and the result holds for all greedoids on $n - 1$ elements. We also assume that e is the first element deleted and contrasted in forming the computation tree, so $T(G)$ looks as in Fig. 1. Suppose the feasible sets $\{A_1, A_2, \ldots, A_d\}$ of G have been ordered so that $e \in A_k$ for $1 \le k \le j$ and $e \notin A_k$ for $j + 1 \le k \le d$. (Note $j \ge 1$, since $\{e\}$ is feasible, and $j < d$, since ϕ is feasible.) $T(G/e)$, by induction, satisfies the theorem, so the leaves of $T(G/e)$ correspond to the feasible sets of G/e, which by definition are all sets of the form $A - \{e\}$, where $e \in A$ and A is feasible in G. Thus, $F_k - \{e\} = A_k - \{e\}$, so $F_k = A_k$ for $1 \le k \le j$ (possibly after reordering these sets). Similarly, $T(G - e)$, by induction, satisfies the theorem, so the leaves of $T(G - e)$ correspond to feasible sets of $G - e$, which by definition are all feasible sets of G which do not contain e. Thus, $m = d$ and $F_k = A_k$ for $j + 1 \le k \le m$ (again after possibly reordering these sets). Since e appears in every feasible set F_k $(1 \le k \le j)$ and e

appears in no feasible set F_k $(j+1 \leq k \leq m)$, these subcomputation trees can be reassembled without repeating any feasible set of G. Thus, we have shown that the leaves of $T(G)$ are in one-to-one correspondence with the feasible sets of G. Finally, $\mathrm{ext}_T(F_k) \subset \sigma(F_k) - F_k$, because if $x \in \mathrm{ext}_T(F_k)$, then $r(F_k \cup x) = r(F_k)$, (or else $\{x\}$ would be feasible in G_k, which has rank 0). Thus, $x \in \sigma(F_k)$, but clearly $x \notin F_k$. This completes the proof.

Antimatroids have been discovered and rediscovered several times. They were introduced by Dilworth (in a lattice-theoretic form) in the 1940s. See [2 or 3] for more background on antimatroids. We now use the above theorem to get an algorithmic characterization of antimatroids.

PROPOSITION 2.4. *Let G be a greedoid and let $T(G)$ be any computation tree for G. Then $\mathrm{ext}_T(F) = \sigma(F) - F$ for all feasible sets F if and only if G is an antimatroid.*

Proof. Let G be an antimatroid, let F be a feasible set of G, and suppose $x \in \sigma(F) - F$. We must show $x \in \mathrm{ext}_T(F)$. (The reverse containment $\mathrm{ext}_T(F) \subseteq \sigma(F) - F$ is given by Proposition 2.3). Let G' be the rank 0 greedoid which labels the leaf of $T(G)$ corresponding to the feasible set F (with the correspondence given in Proposition 2.3). Clearly x was not contracted along the path from the root (labeled by G) to G' (since, by Proposition 2.3, $x \in F$ if and only if x is contracted along the path). If x was deleted along this path, let F' be the set of elements that were contracted prior to the deletion of x. Then $F' \cup \{x\}$ must be feasible (since there is a path in $T(G)$ in which x is contracted at this stage, and all subsequent elements acted on are deleted). Then, since G is an antimatroid, we must have $F \cup \{x\}$ feasible (by 1.4(c)). But now $x \notin \sigma(F)$, which is a contradiction. Thus, x was neither deleted nor contracted along the path from the root to the leaf corresponding to F, so x is externally active for F. Therefore, $\mathrm{ext}_T(F) = \sigma(F) - F$ for all feasible sets F.

Now suppose $\mathrm{ext}_T(F) = \sigma(F) - F$ for all feasible sets F. We will show G is an antimatroid by using the local union property which characterizes antimatroids (1.4(b)). Let $F, F \cup \{x\}$ and $F \cup \{y\}$ all be feasible sets. We must show $F \cup \{x, y\}$ is also feasible. If $F \cup \{x, y\}$ is not feasible, then $x \in \sigma(F \cup \{y\})$ and $y \in \sigma(F \cup \{x\})$. Consider the two paths P_x and P_y from the root of $T(G)$ which terminate in the leaves corresponding to $F \cup \{x\}$ and $F \cup \{y\}$, respectively. Since P_x and P_y coincide at the root, they must diverge at some vertex of $T(G)$ which we assume has label G'. Then the two children of this vertex must be labeled G'/x and $G' - x$ or they must be labeled G'/y and $G' - y$, because x and y are the only elements where these two feasible sets differ. Without loss of generality, we may assume the two children are labeled G'/x and $G' - x$. Then x is deleted along the path P_y; since $\mathrm{ext}_T(F \cup \{y\})$ is the set of elements which were neither deleted nor contracted, $x \notin \mathrm{ext}_T(F \cup \{y\}) = \sigma(F \cup \{y\}) - (F \cup \{y\})$, which contradicts $x \in \sigma(F \cup \{y\})$. This completes the proof.

We do not consider complexity issues associated with implementing the algorithm suggested by 2.4 here. Instead, we turn our attention to the first main application of Propositions 2.3 and 2.4. The next theorem is a direct generalization of Lemma 8.6.1 of [3], which gives a partition of the set **S** of all spanning sets of a greedoid G into Boolean intervals in which each basis of G is the minimum element of some interval.

THEOREM 2.5. *Let $G = (E, \mathscr{F})$ be a greedoid with feasible sets $\mathscr{F} = \{F_1, F_2, \dots, F_m\}$. Then there is a collection of subsets $H_k \subseteq \sigma(F_k)$ $(1 \le k \le m)$ such that the intervals of the form $[F_k, H_k]$ $(1 \le k \le m)$ partition 2^E. Furthermore, this partition is unique if and only if G is an antimatroid. In this case, $H_k = \sigma(F_k)$ for all k.*

Proof. The sets H_k which form the maximum elements in the interval partition of 2^E can be obtained directly from a computation tree $T(G)$: $H_k = F_k \cup \mathrm{ext}_T(F_k)$. The result now follows from 2.3 and 2.4.

Since different computation trees can give rise to different external activities, we can also get different interval partitions of 2^E when G is not an antimatroid. For instance, in Example 2.2, the interval $[\{a, c\}, \{a, b, c\}]$ appears in the partition induced by $T(G)$, but does not in $T'(G)$.

We now define a notion of internal activity for bases with respect to a computation tree $T(G)$. For a basis B of a greedoid G, $e \in B$ is *internally active* with respect to the computation tree $T(G)$ if e is a feasible coloop in some minor occurring in the unique path from the root to the leaf corresponding to B. The set $\mathrm{int}_T(B)$ denotes the set of internally active elements of B for the tree $T(G)$.

Internal and external activity for matroids are usually defined via a total order Ω of the ground set E. In particular, $e \notin B$ is externally active if e is the least element in the unique (basic) circuit contained in $B \cup \{e\}$ and $e \in B$ is internally active if e is the least element in the unique (basic) bond contained in $(E - B) \cup \{e\}$. Let $i_\Omega(B)$ and $e_\Omega(B)$ denote the set of internally and externally active elements for the basis B with respect to Ω. (See [1] for more details on matroid activity.) The next result shows our definition of internal and external activity for a greedoid based on the computation tree $T(G)$ coincides with this definition when G is a matroid. We omit the proof.

PROPOSITION 2.6. *Let G be a matroid with a total order Ω of the ground set E. Then there is a computation tree $T(G)$ such that $i_\Omega(B) = \mathrm{int}_T(B)$ and $e_\Omega(B) = \mathrm{ext}_T(B)$ for all bases B of G.*

The computation tree $T(G)$ in 2.6 is formed by using Ω to operate on the elements of G in reverse order, starting with the largest nonloop (= feasible element), so that the sequence of elements deleted and contracted is the same in every path from the root to the leaves. Adhering to a fixed order is always possible for a matroid, although Example 3.3 below shows this is not true for a general greedoid.

A theorem of Crapo shows how to get an interval partition of 2^E when G is a matroid such that each interval contains a unique basis of G. (See [1 or 9].) We now generalize this result to greedoids.

PROPOSITION 2.7. *Let G be a greedoid with a computation tree $T_B(G)$ and bases $B_1, B_2, \ldots, B_m$. Then the intervals of the form $[B_k - \mathrm{int}_T(B_k), B_k \cup \mathrm{ext}_T(B_k)]$ $(1 \le k \le m)$ partition 2^E.*

Proof. We use induction (as in the proof of 2.3): if $r(G) = 0$, then the partition is trivial.

Let $\{e\}$ be feasible in G. To simplify notation, we write B'_k and B''_k for the bases of G/e and $G - e$, respectively, and let $A_k = B_k - \mathrm{int}_T(B_k)$, $C_k = B_k \cup \mathrm{ext}_T(B_k)$, $A'_k = B'_k - \mathrm{int}_{T/e}(B'_k)$, $A''_k = B''_k - \mathrm{int}_{T-e}(B''_k)$, $C'_k = B'_k \cup \mathrm{ext}_{T/e}(B'_k)$, and $C''_k = B''_k \cup \mathrm{ext}_{T-e}(B''_k)$ (where T/e is the subcomputation tree rooted at G/e and $T - e$ is the subcomputation tree rooted at $G - e$).

If $r(G) > 0$ and e is a feasible coloop of G, then by induction on G/e, we obtain an interval partition $[A'_k, C'_k]$ $(1 \le k \le m)$ of $2^{E-\{e\}}$. Then, for all k, $B'_k = B_k - \{e\}$, $\mathrm{int}_{T/e}(B'_k) = \mathrm{int}_T(B_k) - \{e\}$, $\mathrm{ext}_{T/e}(B'_k) = \mathrm{ext}_T(B_k)$, so $A_k = A'_k$ and $C_k = C'_k \cup \{e\}$ and this gives the desired partition of 2^E.

Now assume $\{e\}$ is feasible and is not a coloop. Assume the bases have been ordered so that $e \in B_k$ for $1 \le k \le j$ and $e \notin B_k$ for $j + 1 \le k \le d$ (where $1 < j < m$ since e is not a coloop). By induction applied to G/e, the intervals of the form $[A'_k, C'_k]$ $(1 \le k \le j)$ partition $2^{E-\{e\}}$. Similarly, the intervals of the form $[A''_k, C''_k]$ $(j + 1 \le k \le m)$ also partition $2^{E-\{e\}}$. In G/e, we find $\mathrm{int}_{T/e}(B'_k) = \mathrm{int}_T(B_k)$ and $\mathrm{ext}_{T/e}(B'_k) = \mathrm{ext}_T(B_k)$; thus $A_k = A'_k \cup \{e\}$ and $C_k = C'_k \cup \{e\}$ (for $1 \le k \le j$). Finally, in $G - e$ we again obtain $\mathrm{int}_{T-e}(B''_k) = \mathrm{int}_T(B_k)$ and $\mathrm{ext}_{T-e}(B''_k) = \mathrm{ext}_T(B_k)$, so $A_k = A''_k$ and $C_k = C''_k$ (for $j + 1 \le k \le m$). This gives the desired partition of 2^E.

The two different interval partitions given in 2.5 and 2.7 represent distinct generalizations of Crapo's interval partition for matroids (Theorem 7.3.6 of [1]; see the notes for Section 7.3 there). They can also both be viewed as distinct generalizations of Lemma 8.6.1 of [3]. When G is a matroid, 2.5 gives a partition of 2^E into intervals of the form $[I, S]$, where I ranges over all independent sets of G and $I \subseteq S \subseteq \bar{I}$.

For greedoids, the partition of 2.5 is more useful than that of 2.7. The interval partition obtained in 2.5 has the property that every subset in a given interval has the same rank; this property allows us to prove Theorem 3.1. For a general greedoid, the partition of 2.7 is not well-behaved in the same sense. We will explore the difference in more detail in the next section.

3. FEASIBLE SETS, ACTIVITIES AND THE TUTTE POLYNOMIAL

We now apply our computation trees to the polynomial $f(G)$.

THEOREM 3.1. *Let $T(G)$ be any computation tree. Then*

$$f(G;t,z) = \sum_{F \text{ feasible}} t^{r(G)-|F|}(z+1)^{|\mathrm{ext}_T(F)|}.$$

Proof. Let $F(G)$ denote the family of feasible sets and $I(F)$ denote the interval $[F, F \cup \mathrm{ext}_T(F)]$. The interval partition of Theorem 2.5 has the property that every subset $S \in I(F)$ has $r(S) = |F|$. Then, we can rewrite the subset expansion of 1.2 as

$$\begin{aligned} f(G;t,z) &= \sum_{s \subseteq E} t^{r(G)-r(S)} z^{|S|-r(S)} = \sum_{F \in F(G)} \sum_{S \in I(F)} t^{r(G)-|F|} z^{|S|-|F|} \\ &= \sum_{F \in F(G)} t^{r(G)-|F|} \sum_{S \in I(F)} z^{|S|-|F|} \\ &= \sum_{F \in F(G)} t^{r(G)-|F|}(z+1)^{|\mathrm{ext}_T(F)|}. \end{aligned}$$

This result gives a way to compute $f(G)$ that is more efficient than using all subsets of E as in 1.2. For example, the computation tree $T(G)$ in Example 2.2 gives $f(G) = t^2(z+1) + t(z+1) + 2(z+1) + t + 1$. A general greedoid will still have an exponential number of feasible sets (as a function of $|E|$); this is expected in view of results of Jaeger, Welsh, and Vertigan [10] which show almost all evaluations of the Tutte polynomial are $\#P$-complete for many well-behaved classes of matroids.

When G is an antimatroid, $\mathrm{ext}_T(F) = \sigma(F) - F$ for any computation tree $T(G)$. In this case, the resulting formula for $f(G)$ from 3.1 is given in Proposition 2.2 of [6]. We also remark that Theorem 3.1 extends Whitney's independent set expansion of the chromatic polynomial of a graph as well as the corresponding expansion of the Tutte polynomial of a matroid. Furthermore, it provides a direct link between the basis activities definition of the one-variable greedoid polynomial $\lambda(G)$ and the two-variable polynomial $f(G)$. We explore this connection in more detail now.

The usual formulation of $\lambda(G)$ involves several steps (see [2 or 3]): First, choose a total order Ω for the ground set E. Now define an ordering on the bases of G lexicographically so that $B_1 < B_2$ provided the least lexicographic feasible permutation of B_1 precedes the least lexicographic feasible permutation of B_2. Now call $x \notin B$ externally active with respect to the order Ω and the basis B if $B < B \cup \{x\} - \{y\}$ for all $y \in B$ such that $B \cup \{x\} - \{y\}$ is a basis of G. (If $B \cup \{x\} - \{y\}$ is not a basis for any $y \in B$, then the condition is vacuously satisfied and x is externally active.) Let $\operatorname{ext}_\Omega(B)$ denote the set of all externally active elements. Then define

$$\lambda(G; u) = \sum_{B \text{ basis}} u^{|\operatorname{ext}_\Omega(B)|}.$$

This notion of external activity for a basis B based on the order Ω (developed in [2]) can be compared to external activity in the computation tree $T(G)$. We now show that these ideas are related by using the order Ω to define a computation tree $T_\Omega(G)$. Given the total order Ω, define $T_\Omega(G)$ recursively as before, replacing 2.1(b) by:

(b′) If e is the first element in the order Ω with $\{e\}$ feasible in G, then recursively form $T_\Omega(G)$ as in Fig. 1 and remove e from the order Ω.

Thus, the order Ω is used to decide which element is deleted and contracted at each step. This is similar to the computation tree based on a total order used in Proposition 2.6.

PROPOSITION 3.2. *Let Ω be a total order of the ground set E of a greedoid G. Then, for all bases B of G,* $\operatorname{ext}_\Omega(B) = \operatorname{ext}_T(B)$ *for the computation tree $T = T_\Omega(G)$.*

Proof. Let e be the first feasible element in Ω and proceed by induction. Then $e \notin \operatorname{ext}_T(B)$ for any basis B of G since e is immediately contracted (for the bases which contained e) or deleted (for the bases which do not contain e) in $T_\Omega(G)$. Thus, $\operatorname{ext}_{T/e}(B - e) = \operatorname{ext}_T(B)$ for all bases B containing e and $\operatorname{ext}_{T-e}(B) = \operatorname{ext}_T(B)$ for the remaining bases of G.

But this relation also holds for $\operatorname{ext}_\Omega(T)$: if $\Omega - e$ is the ordering obtained by removing e from Ω, then $\operatorname{ext}_{\Omega-e}(B - e) = \operatorname{ext}_\Omega(B)$ for all bases B containing e and $\operatorname{ext}_{\Omega-e}(B) = \operatorname{ext}_\Omega(B)$ for all bases not containing e. This follows from the proof of Theorem 6.4(i) of [2]. By induction, $\operatorname{ext}_{\Omega-e}(B - e) = \operatorname{ext}_{T/e}(B - e)$ for all bases containing e and $\operatorname{ext}_{\Omega-e}(B) = \operatorname{ext}_{T-e}(B)$ for all bases not containing e. Thus, for the bases containing e, $\operatorname{ext}_T(B) = \operatorname{ext}_{T/e}(B - e) = \operatorname{ext}_{\Omega-e}(B - e) = \operatorname{ext}_\Omega(B)$, and for the bases not containing e, $\operatorname{ext}_T(B) = \operatorname{ext}_{T-e}(B) = \operatorname{ext}_{\Omega-e}(B) = \operatorname{ext}_\Omega(B)$.

This completes the proof.

Thus, Theorem 3.1 is a direct generalization of the basis activities definition of $\lambda(G)$. The order Ω gives a recipe for creating $T_\Omega(G)$, but the order may vary for different paths of the computation tree (unlike the situation for matroids). The next example shows how this can occur.

EXAMPLE 3.3. Let G be the directed branching greedoid associated with the rooted digraph of Fig. 3.

Let $B_1 = \{a, c, d, e\}$ and let $B_2 = \{b, c, d, f\}$. Then in any computation tree $T(G)$ associated to G, c must be contracted before d along the path corresponding to the basis B_1, and d must be contracted before c along the path corresponding to the basis B_2. Thus, no single total order can be followed along each path in $T(G)$.

We close this section by interpreting the difference between the two interval partitions given in 2.5 and 2.7 in terms of greedoid polynomials. Let $T_B(G)$ be the subtree obtained from a computation tree $T(G)$ by simply removing all vertices of $T(G)$ in which a coloop was deleted. Equivalently, we can view $T_B(G)$ as the subtree formed by all the paths in $T(G)$ which emanate from the root and terminate at a leaf which corresponds to a basis. Then $T_B(G)$ is the tree which is used to compute the one-variable greedoid polynomial $\lambda(G)$. See Fig. 6.3 of [2] for an example. We note that the inductive proof of 2.7 gives a recursive algorithm for creating such a tree $T_B(G)$.

Imitating Tutte's original approach, it is tempting to create a two-variable greedoid Tutte polynomial as follows:

Let $T(G)$ be a computation tree for a greedoid G and let $B(G)$ denote the family of bases of G. Then define

$$h_T(G; x, y) = \sum_{B \in B(G)} x^{|\mathrm{int}_T(B)|} y^{|\mathrm{ext}_T(B)|}.$$

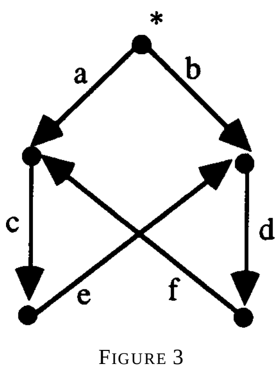

FIGURE 3

Unfortunately, internal activity is not well-behaved and this polynomial depends in a crucial way on $T(G)$. For example, consider the directed branching greedoid associated with the rooted digraph of Fig. 4. Using a subcomputation tree T_1 based on the order $a < b < c < d < e < f$ gives $h_1(G) = x^3y + 2x^2y^2 + xy^2 + y^3$, while using a subcomputation tree T_2 based on the order $b < a < c < d < e < f$ gives $h_2(G) = x^3y^2 + x^2y + 2xy^2 + y^3$.

By 3.1, it is easy to see that $h_T(G; 1, y) = f(G; 0, y - 1) = \lambda(G; y)$. The difference between internal and external activity in greedoids is (indirectly) related to the fact that the primal simplicial complex of a greedoid (which is the heredity closure of the family of bases of G) is not shellable in general, while the dual complex (the hereditary closure of the family of basis complements) is always shellable. The discussion following Theorem 8.6.7 of [3] contains more information concerning the relationship between shellability and $\lambda(G)$.

4. DIRECTED BRANCHING GREEDOIDS: AN APPLICATION

When G is a matroid with k loops, then $(z + 1)^k$ divides $f(G; t, z)$; conversely, when $(z + 1)^k$ divides $f(G; t, z)$, G must have k loops. The first implication is true for greedoids (see 4.1(d) below); the second is not. We explore this in detail for directed branching greedoids, i.e., the greedoids associated with rooted digraphs.

We first list some basic results. If D is rooted digraph with (directed) edge set E, the *directed branching greedoid* $G(D) = (E, \mathscr{F})$ is defined as follows: a set $T \subseteq E$ is feasible in $G(D)$ if T forms a rooted arborescence of D, i.e., a directed tree in which all of the edges of T are directed "away" from the root. The next result gives graph theoretic interpretations

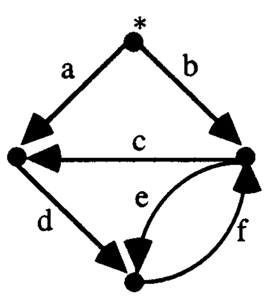

FIGURE 4

for some of the greedoid concepts we have used. See [12] for more discussion.

LEMMA 4.1. *Let $G(D)$ be the directed branching greedoid associated with the rooted digraph D with root vertex *.*

(a) *A directed edge e with initial vertex v and terminal vertex w is a greedoid loop in $G(D)$ if and only if w lies on every directed path from * to v.*

(b) *If every vertex of D is accessible from * via some directed path, then a subdigraph D' will span (in the greedoid sense) $G(D)$ if and only if, for every vertex v of D, there is a path from * to v using only edges of D'. (Such a subdigraph D' is said to be* spanning *in D.)*

(c) *If D has no greedoid loops and D' is spanning in D, then D' has no greedoid loops.*

(d) *If D has m greedoid loops and D' is the digraph obtained from D by removing these m loops, then $f(G(D)) = (z+1)^m f(G(D'))$.*

Remarks. 4.1(a) includes "classic" loops (in which the initial and terminal vertices coincide) as well as edges in which the initial vertex is not accessible from *. The condition in 4.1(b) about all vertices of D being accessible from * makes sense for the purpose of analyzing $f(G(D))$. In particular, isolated vertices are irrelevant (since the ground set for the greedoid is the edge set); thus we always assume D has no isolated vertices. Further, if D has an inaccessible vertex v (from *), any edge incident to v will be a greedoid loop (as 4.1(a) is vacuously satisfied). Since greedoid loops behave in a predictable way with respect to the Tutte polynomial (4.1(d)), we will concentrate on digraphs D in which there are no isolated vertices or greedoid loops.

We now turn our attention to the main results of this section. When G is the directed branching greedoid associated with a rooted digraph, we obtain (Theorem 4.2) a graph theoretic interpretation for the highest power of $(z+1)$ which divides $f(G)$. This result generalizes Theorem 3 of [12], where it was shown that when G is the directed branching greedoid associated with a rooted digraph D having no greedoid loops, then D has a directed cycle if and only if $(z+1)$ divides $f(G)$.

THEOREM 4.2. *Let G $(= G(D))$ be the directed branching greedoid associated with a rooted digraph D with no greedoid loops or isolated vertices. If $f(G; t, z) = (z+1)^k f_1(t, z)$, where $(z+1)$ does not divide $f_1(t, z)$, then k is the minimum number of edges that need to be removed from D to leave a spanning, acyclic directed graph.*

Proof. Assume D is a rooted directed graph with no greedoid loops, and $f(G(D); t, z) = (z+1)^k f_1(t, z)$, where $(z+1)$ does not divide $f_1(t, z)$.

Let m be the minimum number of edges whose removal leaves an acyclic, spanning subdigraph.

Part 1. $m \leq k$. We will construct a set of k edges in D whose removal leaves an acyclic, spanning digraph. Let $T(G)$ be a computation tree for G and let e be the first edge deleted and contracted in $T(G)$. Then the multiplicity of $(z+1)$ in $f(G; t, z)$ is exactly equal to the smaller multiplicity of $(z+1)$ in $f(G/e; t, z)$ or $f(G-e; t, z)$ (Lemma 1 of [12]). Applying this repeatedly, we can form a path P in $T(G)$ by choosing either H/e' or $H-e'$ for every subgreedoid H encountered along the way, according to which of $f(H/e')$ or $f(H-e')$ contains exactly k factors of $(z+1)$. If both $f(H/e')$ and $f(H-e')$ contain exactly k factors of $(z+1)$, choose H/e' to form the paths. Thus, our path P in $T(G)$ begins at the root, ends at a leaf, and has the property that $(z+1)$ has multiplicity k in $f(H)$ for every subgreedoid H on P. Let B be the feasible set which labels the leaf of P. Then by Theorem 3.1, $|\mathrm{ext}_T(B)| = k$.

Now let D' be the rooted digraph obtained from D after the edges corresponding to these k elements of G are removed from D (i.e., after $\mathrm{ext}_T(B)$ is removed from G) and let G' $(= G(D'))$ be the directed branching greedoid associated with D'. We form a computation tree $T(G')$ which has the property that the sequence of deletions and contractions along one path P' will be precisely the same as the sequence along P in $T(G)$ (the rest of $T(G')$ can be formed arbitrarily). The minor which labels the leaf of the path P' is empty (i.e., the corresponding feasible set of G' has no externally active elements), since every edge was either deleted or contracted. Thus, the polynomial for that minor has no factors of $(z+1)$, so the polynomial $f(G')$ has no factors of $(z+1)$ by Theorem 3.1. By Lemma 4.1(d), D' has no greedoid loops. Thus, we can apply Theorem 3 of [12] to conclude that D' is acyclic.

It remains to show that D' spans D. Suppose that this is not the case. Then there is a vertex v which is not reachable from * in D'. Since v is reachable from * in D, there must be consecutive nodes labeled H and $H-e$ (for some edge e) along the path P in the computation tree such that there is a path from * to v in the digraph corresponding to H, but no such path in the digraph corresponding to $H-e$. (Note that the edge e must be deleted along P.) Let w be the terminal vertex of e and let e' be an edge in a path from * to v (in the digraph corresponding to H) whose initial vertex is w. Then the set $B \cup \{e\}$ is feasible and $e' \in \mathrm{ext}_T(B)$.

Without loss of generality, we may assume that the computation tree $T(G)$ contains a path P' which agrees with the path P from the root to the node labeled by the minor H, then selects the node labeled by H/e, and then uses precisely the same sequence of deletions and contractions which occurred along the path P. Further, the path P' must give rise to a higher number of externally active elements at its node, by the way the path P was chosen. To see why it is always possible to find such a path P', we must show that if f is feasible in a minor H'_d of $H - e$, then f is also feasible in the corresponding minor H'_c of H/e. But this is clear, since there must be a path from * which includes the edge f in $H - e$ and this path avoids the vertex w (as $e' \in \mathrm{ext}_T(B)$). Thus the same path including f exists in H/e, and f will be feasible in H'_c.

Since * and w coincide in H/e, the edge e' must eventually be deleted or contracted in the computation tree. Thus $e' \notin \mathrm{ext}_T(B \cup \{e\})$. We claim that $|\mathrm{ext}_T(B \cup \{e\})| < k$, producing a term $t^a(z+1)^b$ in $f(G; t, z)$ with $b < k$ (by Theorem 3.1), which would contradict the minimality of k. To show this, we will prove that $\mathrm{ext}_T(B \cup \{e\}) \subseteq \mathrm{ext}_T(B)$.

Suppose $d \in \mathrm{ext}_T(B \cup \{e\})$ and let $H(d)$ denote the minor labeling a node on the path P' in which d first becomes a greedoid loop. There are three possibilities for where $H(d)$ can appear along P'.

Case 1. $H(d)$ labels a node which P' shares with P. Clearly d will be a greedoid loop in all of the descendants of $H(d)$, so $d \in \mathrm{ext}_T(B)$.

Case 2. $H(d) = H/e$. Then either d and e are parallel edges or d and e must share the same terminal vertex w, from [12]. But d cannot be parallel to e in the digraph corresponding to H, since there is no path from * to v in the digraph corresponding to $H - e$. Further, since v is not reachable from * in the digraph corresponding to $H - e$, any edge $d \neq e$ having terminal vertex w will be a greedoid loop in $H - e$, i.e., $d \in \mathrm{ext}_T(B)$.

Case 3. $H(d)$ is a proper minor H/e. If d is not a greedoid loop in $H - e$, then by 4.1(a) there is at least one feasible path from * to the terminal vertex of d in $H - e$. Furthermore, as before, no such path can pass through the vertex w. Thus each such path in $H - e$ gives rise to a corresponding path in H/e. The sequence of deletions and contractions which produce $H(d)$ must destroy all these paths in H/e; hence the same sequence of deletions and contractions will also destroy all such paths in a corresponding minor of $H - e$, i.e., $d \in \mathrm{ext}_T(B)$.

We have shown $\mathrm{ext}_T(B \cup \{e\}) \subseteq \mathrm{ext}_T(B) - \{e'\}$, so $|\mathrm{ext}_T(B \cup \{e'\})| < k$. This gives us the desired contradiction.

Thus we have produced k edges whose removal leaves an acyclic, spanning subgraph. This completes this part of the proof.

Part 2. $m \geq k$. We reverse the process used in the first part of the proof. First choose a minimal set of m edges whose removal leaves an acyclic, spanning digraph D'. Let $T(G')$ be any computation tree for the directed branching greedoid G' $(= G(D'))$. Then, by Theorem 3 of [12], $(z+1)$ does not divide $f(G')$, so, by Theorem 3.1, there is a path P' in $T(G')$ from the root to a leaf labeled by an empty minor, i.e., a feasible set with no external activity. We now create a computation tree for $G(D)$.

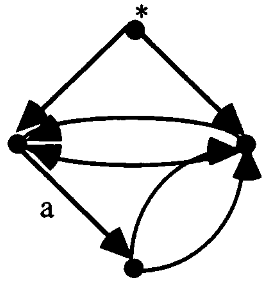

FIGURE 5

Consider the sequence of deletions and contractions along P' in $T(G')$ and form a computation tree $T(G)$ with a path P that has precisely the same sequence of deletions and contractions as P' does (with the rest of $T(G)$ completed arbitrarily). (Again, this is always possible, since adding the m edges to D' will not make an edge which was feasible in D' become not feasible in D.) Suppose this process terminates at a vertex of $T(G)$ labeled by the minor H. By Theorem 3.1 and the construction of the path P, either H or one of its descendants will contribute the term $t^a(z+1)^b$ to the polynomial $f(G)$ for some $b \leq m$. But $k \leq b$ since $(z+1)^k$ divides each term of $f(G)$, so $k \leq m$. This completes the proof.

Theorem 4.2 generalizes a similar result which holds for the polynomial $\lambda(G)$. See the discussion concerning directed branching greedoids following Proposition 8.6.3 in [3].

The necessity of the spanning condition in 4.2 can be seen in the following example.

EXAMPLE 4.3. Let D be the rooted digraph of Fig. 5. Then $(z+1)^2$ divides the Tutte polynomial, so the minimum number of edges which can be removed to leave an acyclic spanning digraph is 2. Removing the edge labeled a leaves an acyclic digraph, but the resulting digraph does not span.

ACKNOWLEDGMENT

We wish to thank the anonymous referee for many helpful suggestions in clarifying the exposition in Section 4 and the proof of Theorem 4.2 and for contributing Example 4.3.

REFERENCES

1. A. Björner, Homology and shellability of matroids and geometric lattices, *in* ''Matroid Applications'' (N. White, Ed.), Encyclopedia of Mathematics and Its Applications, Vol. 40, pp. 226–283, Cambridge Univ. Press, London, 1992.
2. A. Björner, B. Korte, and L. Lovász, Homotopy properties of greedoids, *Adv. Appl. Math.* **6** (1985), 447–494.
3. A. Björner and G. Ziegler, Introduction to Greedoids, *in* ''Matroid Applications'' (N. White, Ed.), Encyclopedia of Mathematics and Its Applications, Vol. 40, pp. 284–357, Cambridge Univ. Press, London, 1992.
4. T. Brylawski and J. Oxley, The Tutte polynomial and its applications, *in* ''Matroid Applications'' (N. White, Ed.), Encyclopedia of Mathematics and Its Applications, Vol. 40, pp. 123–225, Cambridge Univ. Press, London, 1992.
5. S. Chaudhary and G. Gordon, Tutte polynomials for trees, *J. Graph. Theory* **15** (1991), 317–331.
6. G. Gordon, A Tutte polynomial for partially ordered sets, *J. Combin. Theory Ser. B* **59** (1993), 132–155.
7. G. Gordon, Series-parallel posets and the Tutte polynomial, *Discrete Math.*, **158** (1996), 63–75.
8. G. Gordon and E. McMahon, A greedoid polynomial which distinguishes rooted arborescences, *Proc. Amer. Math. Soc.* **107** (1989), 287–298.
9. G. Gordon and L. Traldi, Generalized activities and the Tutte polynomial, *Discrete Math.* **85** (1989), 167–176.
10. F. Jaeger, D. Vertigan, and D. J. A. Welsh, On the computational complexity of the Jones and Tutte polynomials, *Math. Proc. Camb. Phil. Soc.* **108** (1990), 35–53.
11. B. Korte, L. Lovász, and R. Schrader, ''Greedoids,'' Springer-Verlag, Berlin, 1991.
12. E. McMahon, On the greedoid polynomial for rooted graphs and rooted digraphs, *J. Graph Theory* **17** (1993), 433–442.