# The Tutte polynomial of a graph, depth-first search, and simplicial complex partitions

Dedicated to Dominique Foata on the occasion of his 60th birthday

Appeared in *Electronic J. Combin. (Foata Festschrift Volume)*, **3** (1996)

Ira M. Gessel
Department of Mathematics
Brandeis University
Waltham, MA 02254-9110

Bruce E. Sagan
Department of Mathematics
Michigan State University
East Lansing, MI 48824-1027

Key Words: Tutte polynomial, simplicial complex, partition, depth-first search, spanning tree, acyclic orientation, inversion, parking function

AMS subject classification (1985): Primary 05C30; Secondary 05C05, 68R10.

**Abstract**

One of the most important numerical quantities that can be computed from a graph $G$ is the two-variable Tutte polynomial. Specializations of the Tutte polynomial count various objects associated with $G$, e.g., subgraphs, spanning trees, acyclic orientations, inversions and parking functions. We show that by partitioning certain simplicial complexes related to $G$ into intervals, one can provide combinatorial demonstrations of these results. One of the primary tools for providing such a partition is depth-first search.

# 1 Introduction and definitions

Tutte defined the polynomial that bears his name as the generating function for two parameters, namely the internal and external activities, associated with the spanning trees of a connected graph. In this paper we will propose a number of new, but related, notions of external activity. We will show in the next six sections that using these other definitions of activity can lead to combinatorial proofs of results about many specializations of $t_G$. These evaluations count subgraphs, acyclic orientations, subdigraphs, inversions and parking functions. The basic idea is to use depth-first search to associate a spanning forest $F$ with each object to be counted. This partitions the simplicial complex of all objects (ordered by inclusion) into intervals, one for each $F$. Every interval turns out to be a Boolean algebra consisting of all ways to add externally active edges to $F$. Expressing the Tutte polynomial in terms of sums over such intervals permits us to extract the necessary combinatorial information.

The idea of using partitions to get information about the Tutte polynomial goes back to Crapo [8] and has also been used by Bari [2], Dawson [9, 10], Gessel and Wang [18], Gordon and Traldi [20], and others. See Björner [4] for a good account of the connection between Tutte polynomials, partitions, and shellability. See Brylawski [5] and his survey with Oxley [6] for the general theory of the Tutte polynomial. Partitioning simplicial complexes into Boolean algebras also has other applications. See, for example, the paper of Garsia and Stanton [15]. Finally, we should mention that Kleitman and Winston [23] have used depth-first search to construct a bijection in a context similar to ours. However, our paper is the first to systematically mine the combination of the partitioning and DFS ideas to obtain a wide range of results.

Let $G$ denote a graph with vertex set $V$. In most of our work (in particular, for depth-first search and its variations) we assume that $V$ is totally ordered. Often we will take $V = \{1, 2, \ldots, n\}$. We will permit $G$'s edge set to have loops and multiple edges, calling two edges with the same endpoints *parallel*. It will be convenient to identify a graph with its edge set and use the notation $G$ for both. All the previous conventions will apply to digraphs as well.

All of our subgraphs are *spanning*, that is, a subgraph of $G$ has the same vertex set as $G$. Thus we identify subgraphs of $G$ with subsets of the edge set. A subgraph of $G$ is a *forest* if it is acyclic; in particular, it must contain no loops or parallel edges. The connected components of a forest are *trees*.

Before defining the Tutte polynomial, $t_G(x, y)$, we make the convention that, except where stated otherwise, $G$ is a connected graph. This is no loss of generality for two reasons. First, if $G$ is disconnected then $t_G(x, y)$ is just the product of the Tutte polynomials of the components of $G$. Furthermore, most of the algorithmic constructions that we will use can be carried out in $G$ by applying them to each component. In the few places where this is not true, we will indicate what modifications need to be made for the general case.

Now suppose we are given $G$ and a total ordering of its edges. Consider a spanning tree $T$ of $G$. An edge $e \in G - T$ is *externally active* if it is the largest edge in the unique cycle contained in $T \cup e$. We let

$$\mathcal{EA}(T) = \text{ set of externally active edges of } T$$

and

$$ea(T) = |\mathcal{EA}(T)|$$

where $|\cdot|$ denotes cardinality. Of course, the set of externally active edges depends on both $G$ and $T$. However, $G$ will always be clear from context. An edge $e \in T$ is *internally active* if it is the largest edge in the unique cocycle contained in $(G - T) \cup e$. (A *cocycle* is a minimal disconnecting subset of $G$.) We let

$$\mathcal{IA}(T) = \text{ set of internally active edges of } T$$

and

$$ia(T) = |\mathcal{IA}(T)|.$$

Tutte [40] then defined his polynomial as

$$t_G(x,y) = \sum_{T \subseteq G} x^{ia(T)} y^{ea(T)} \tag{1}$$

where the sum is over all spanning trees $T$ of $G$. Tutte showed that $t_G$ is well-defined, i.e., independent of the ordering of the edges of $G$. Henceforth, we will not assume that the edges of $G$ are ordered unless it is explicitly stated.

We end this section by reviewing the notion of depth-first search, which we abbreviate to DFS. Given a graph $H$ with vertex set $V$, we will use the following algorithm to create the DFS forest $F$ of $H$.

- DFS1 Let $F := \emptyset$.
- DFS2 Let $v$ be the least unvisited vertex in $V$. Mark $v$ as visited.
- DFS3 Pick some unvisited vertex $u$ adjacent to $v$ by an edge $e$ if such a vertex exists. Mark $u$ as visited and set $v := u, F := F \cup e$. Repeat this step until $v$ has no unvisited neighbors.
- DFS4 If there is a visited vertex with unvisited neighbors, let $v$ be the most recently visited such vertex and go to DFS3. Otherwise, go to DFS2 and repeat the process until all vertices are visited.

The DFS variants that we will introduce in the next sections are all constructed by specifying the vertex $u$ and the edge $e$ in step DFS3.

## 2 Subgraphs

Let $G$ be a graph with $|V| = n$, and let $H$ be a subgraph. We denote by $c(H)$ the number of components of $H$. We introduce two useful invariants associated with $H$, namely

$$\sigma(H) = c(H) - c(G) = c(H) - 1 \tag{2}$$

and

$$\sigma^*(H) = |H| - |V| + c(H) = |H| - n + c(H) \tag{3}$$

where we recall that $|H|$ denotes the number of edges in $H$. These quantities are naturally associated with the *cycle matroid of* $G$ whose independent sets consist of all spanning forests $F$ of $G$. The *rank* of $H \subseteq G$ in this matroid is

$$\begin{aligned} \rho(H) &= \max\{|F| : F \subseteq H \text{ with } F \text{ a spanning forest of } G\} \\ &= |V| - c(H) \end{aligned}$$

and the *corank* is

$$\begin{aligned} \rho^*(H) &= \max\{|C| : C \subseteq H \text{ with } G - C \text{ connected}\} \\ &= |H| - c(G - H) + 1. \end{aligned}$$

Thus

$$\sigma(H) = \rho(G) - \rho(H)$$

and

$$\sigma^*(H) = \rho^*(G) - \rho^*(G - H).$$

It is well known [4] that one evaluation of the Tutte polynomial is the generating function for subgraphs of $G$ with respect to the invariants $\sigma$ and $\sigma^*$.

**Theorem 2.1** *We have*

$$t_G(1 + x, 1 + y) = \sum_{H \subseteq G} x^{\sigma(H)} y^{\sigma^*(H)} \tag{4}$$

*where the sum is over all subgraphs* $H$ *of* $G$. ■

The version of depth-first search that will be useful in connection with subgraphs $H \subseteq G$ is *greatest-neighbor* DFS. Each time we perform DFS3 we visit the unvisited neighbor with largest label first. (However, we still always start the search at the vertex with smallest label. One of the reasons for these conventions is to coincide with those used later when discussing inversions in trees.) We also assume that each set of parallel edges is totally ordered and that we always take

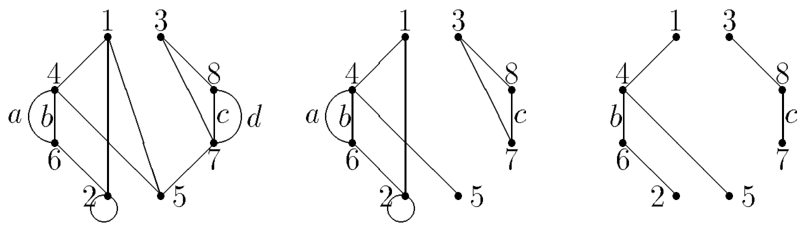


Figure 1: A greatest-neighbor DFS forest $F$ for $H \subseteq G$

the largest edge connecting two vertices. If $H$ has greatest-neighbor DFS forest $F$, then we write $\mathcal{F}^+(H) = F$ which we will often abbreviate to $\mathcal{F}(H) = F$. It is convenient to root each tree of $\mathcal{F}(H)$ at its least vertex. We also note that $\mathcal{F}(H)$ does not depend on the edges of $G - H$ and that $c(\mathcal{F}(H)) = c(H)$.

By way of illustration, suppose we have the graph $G$ in Figure 1. The two sets of parallel edges are ordered lexicographically. Then the given subgraph $H$ has greatest-neighbor forest $F$ with components rooted at vertices 1 and 3.

Now given a spanning forest $F \subseteq G$ let us say that an edge $e \in G - F$ is *(greatest-neighbor) externally active* if

$$\mathcal{F}(F \cup e) = F.$$

We write $\mathcal{E}^+(F)$, or simply $\mathcal{E}(F)$, for the set of greatest-neighbor externally active edges. In our previous example, edges $\{1,2\}, \{3,7\}, \{2,2\}$ and $a = \{4,6\}$ are externally active while $\{1,5\}, \{5,7\}$ and $d = \{7,8\}$ are not.

The next result follows easily from the definitions. In it, $\uplus$ stands for disjoint union.

**Proposition 2.2** *If $H$ is any subgraph and $F$ is any spanning forest of $G$ then*

$$\mathcal{F}(H) = F \quad \Longleftrightarrow \quad F \subseteq H \subseteq F \uplus \mathcal{E}(F). \ \blacksquare$$

Thus the intervals $[F, F \uplus \mathcal{E}(F)]$ partition the simplicial complex of all subgraphs of $G$ into Boolean algebras, one for each spanning forest.

To turn this proposition into an enumerative result, note that if $\mathcal{F}(H) = F$ then $c(H) = c(F)$ so $\sigma(H) = \sigma(F) = c(F) - 1$ and $\sigma^*(H) = |H| - |F| = |H \cap \mathcal{E}(F)|$. Thus, if we fix a forest $F$ and sum over the corresponding interval

$$\sum_{H : \mathcal{F}(H) = F} x^{\sigma(H)} y^{\sigma^*(H)} \quad = \quad x^{\sigma(F)} \sum_{A \subseteq \mathcal{E}(F)} y^{|A|}$$

$$= \quad x^{\sigma(F)}(1+y)^{|\mathcal{E}(F)|}.$$

Summing over all forests $F$, we have

$$\sum_{H \subseteq G} x^{\sigma(H)} y^{\sigma^*(H)} = \sum_{F \subseteq G} x^{\sigma(F)}(1+y)^{|\mathcal{E}(F)|}.$$

But from Theorem 2.1, we know that the left-hand side is just $t_G(1+x, 1+y)$. Thus, changing $y$ to $y-1$, we have

$$t_G(1+x, y) = \sum_{F \subseteq G} x^{\sigma(F)} y^{|\mathcal{E}(F)|}$$

or

$$x \; t_G(1+x, y) = \sum_{F \subseteq G} x^{c(F)} y^{|\mathcal{E}(F)|}, \tag{5}$$

an equation that will be useful in the future. Note that if $G$ were allowed to be disconnected then the factor of $x$ on the left of this equation would be replaced with $x^{c(G)}$.

We will now give a characterization of the edges in $\mathcal{E}(F)$ that will allow us to mine more combinatorial information from equation (5). Consider a tree $T$ in $F$ rooted at its smallest vertex. Then we will use all the usual family tree conventions when talking about vertices of $T$ (parent, child, and so on). Also, we call a pair of vertices $(u, v)$ in $T$ an *inversion* (respectively *non-inversion*) if $u$ is an ancestor of $v$ and $u > v$ (respectively $u < v$). Finally, a *cross edge* is $e = \{u, v\}$ where $u$ is not a descendant of $v$ and vice-versa.

**Lemma 2.3** *Suppose $G$ is a graph with spanning forest $F$ and $e \in G - F$. Then $e \in \mathcal{E}(F)$ if and only if $e$ is of one of the following types:*

1. *$e = \{u, v\}$ where $v$ is a descendant of $u$, and $(w, v)$ is an inversion where $w$ is the child of $u$ on the unique $u$-$v$ path in $F$, or*

2. *$e < f$ where $f \in F$ is an edge with the same endpoints as $e$, or*

3. *$e$ is a loop.*

Figure 2 shows a schematic diagram of an externally active edge corresponding to an inversion. For a concrete example, see Figure 1 where edges $\{1, 2\}$ and $\{3, 7\}$ are of type 1, edge $a = \{4, 6\}$ is of type 2 and $\{2, 2\}$ is of type 3.

**Proof of Lemma 2.3.** It suffices to show that $\mathcal{F}(F \cup e)$ does not contain $e$ if and only if $e$ is one of these three types. If $e$ is of type 1, then DFS will reach $u$ before $v$. But since we are using greatest-neighbors, the search will continue to

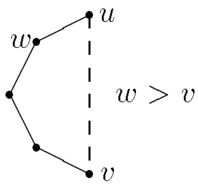


Figure 2: The inversion case of Lemma 2.3

$w$ regardless of the presence of $\{u, v\}$. By the time the search reaches $v$, $u$ has already been visited and so $e$ cannot be used in that direction either. The parallel edge case follows from using greatest edges, and loops are never in forests.

For the converse, suppose that $e$ is not one of the three types. Then $e$ must be of the form

i. $e = \{u, v\}$ where $v$ is a descendant of $u$ and $(w, v)$ is a non-inversion, where $w$ is the child of $u$ on the unique $u$-$v$ path in $F$, or

ii. $e > f$ where $f \in F$ is an edge with the same endpoints as $e$, or

iii. $e$ is a cross edge.

In the first two cases, the greatest-neighbor search will be forced to traverse $e$ the first time it is encountered. So $e \in \mathcal{F}(F \cup e)$. In the third case, suppose $e = \{u, v\}$ and that $u$ is searched first in $F$. Then $v$ would eventually be visited as a neighbor of $u$ in $F \cup e$. Again, this forces $e \in \mathcal{F}(F \cup e)$. ■

# 3 Orientations

If $G$ is a graph, then an *orientation* $\mathcal{O}$ *of* $G$ is a digraph obtained by assigning one of the two possible directions to each edge of $G$. If $e = \{u, v\}$ is an edge then the corresponding arc will be denoted $\vec{e}$ with possible directions $\vec{e} = uv$ or $\vec{e} = vu$. For enumerative purposes, we also consider each loop to have two possible directions. A *suborientation of* $G$ is an orientation of a subgraph of $G$. A digraph is *acyclic* if it contains no directed cycles. Loops and oppositely directed parallel edges are considered cycles.

We can use the Tutte polynomial and DFS to count acyclic suborientations of $G$. Given any digraph $D$, we use *least-neighbor search,* which goes to the smallest vertex at each step. If there are parallel arcs between the two vertices, they are

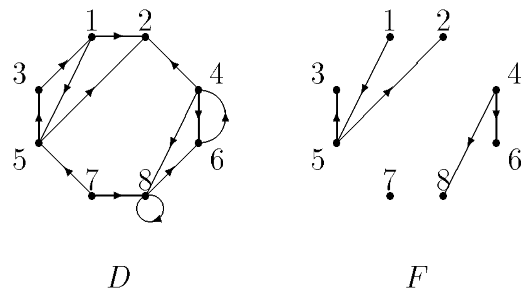


Figure 3: A digraph $D$ and forest $F \subseteq D$

ordered and the smallest one is taken. (In DFS2 we still always start at the least unvisited vertex.) Of course, we only traverse arcs in the proper direction. If $D$ has least-neighbor DFS forest $F$, then we write $\vec{\mathcal{F}}^{-}(D) = F$.

The trees generated by the least-neighbor search of $D$ are related to certain components of the digraph. We call a digraph *initially connected* if there is a directed path from the smallest vertex to any other. An arbitrary digraph can be decomposed into *initial components* as follows. The first component contains all vertices reachable by a directed path from the least vertex. Remove these vertices and the second component will contain vertices reachable from the smallest vertex that remains, etc. The digraph in Figure 3 has three initially connected components, namely the subdigraphs induced by the vertex sets $\{1, 2, 3, 5\}$, $\{4, 6, 8\}$ and $\{7\}$. Note that all arcs between two such components are directed from the later to the earlier component. In general, we say that an arc $uv$ is directed *later to earlier* if $u$ is visited after $v$ in DFS.

We will write $c(D)$ for the number of initial components of $D$. Notice that if $D$ has DFS forest $F$, then $c(D) = c(F)$. Also, $c(F)$ coincides with the number of components of $F$ considered as an undirected graph (sometimes called the *weak components*).

Given a digraph $D$ containing a forest $F$, the arcs $uv \in D - F$ can be partitioned into four types:

- *Forward arcs* where $u$ is an ancestor of $v$,
- *Backward arcs* where $u$ is a descendant of $v$,
- *Cross arcs* where $u$ is neither an ancestor nor a descendant of $v$, and
- *Loops.*

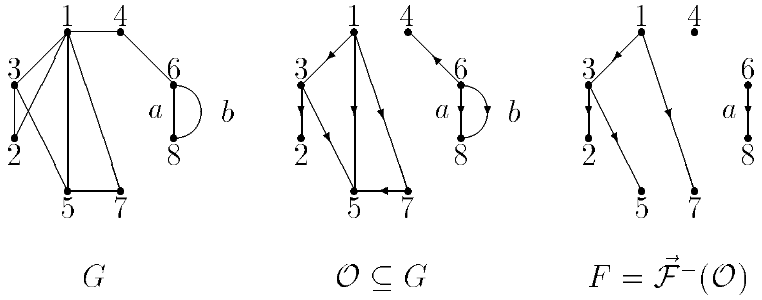


Figure 4: A graph $G$, suborientation $\mathcal{O}$, and least-neighbor forest $F$

For example, with respect to the digraph $D$ and subforest $F$ of Figure 3 we have forward arc 12; backward arcs 31, 64; cross arcs 42, 75, 78, 86; and a loop at the vertex 8. It will be convenient to keep this partition in mind in the future.

Now suppose $G$ is a graph and $F \subseteq G$ is a forest. Consider $F$ to be a suborientation of $G$, where each edge is directed away from the root of its tree. Let $\vec{\mathcal{E}}^-(F)$ be the set of all orientations $\vec{e}$ of edges in $G - F$ such that

1. $F \cup \vec{e}$ is acyclic, and
2. $\vec{\mathcal{F}}^-(F \cup \vec{e}\,) = F$.

We call the arcs in $\vec{\mathcal{E}}^-(F)$ *(directed) least-neighbor externally active.* Note that by the condition 1, $\vec{\mathcal{E}}^-(F)$ never contains any loops. Also, $\vec{\mathcal{E}}^-(F)$ contains at most one orientation of each edge of $G$ which is not a loop: if $e = \{u, v\}$ with $u$ an ancestor or descendant of $v$ then $\vec{e}$ must be oriented forward by condition 1; if $e$ is a cross edge then it must be directed later to earlier by condition 2. If we consider the graph $G$ and suborientation $\mathcal{O}$ in Figure 4, then the arcs in $\vec{\mathcal{E}}^-(F)$ are 15, 41, 64, 75 and $b$ oriented in the direction 68.

The analog of Proposition 2.2 in this setting is as follows.

**Proposition 3.1** *If $\mathcal{O}$ is any suborientation and $F$ is any spanning forest of $G$ then*

$$\mathcal{O} \text{ is acyclic and } \vec{\mathcal{F}}^-(\mathcal{O}) = F \quad \Longleftrightarrow \quad F \subseteq \mathcal{O} \subseteq F \uplus \vec{\mathcal{E}}^-(F).$$

**Proof.** First suppose that $\mathcal{O}$ is acyclic and that $\vec{\mathcal{F}}^-(\mathcal{O}) = F$. Then clearly $F \subseteq \mathcal{O}$. If $\mathcal{O} \not\subseteq F \uplus \vec{\mathcal{E}}^-(F)$, then some $\vec{e} \in \mathcal{O}$ must violate condition 2 (since $\mathcal{O}$ is acyclic by assumption). But this implies $\vec{\mathcal{F}}^-(\mathcal{O}) \neq F$, a contradiction.

For the other direction, condition 2 implies $\vec{\mathcal{F}}^-(\mathcal{O}) = F$. To verify acyclicity, suppose to the contrary that $v_1, v_2, \ldots, v_k, v_1$ is a directed cycle in $\mathcal{O}$. If all the

$v_i$ are on the same path from the root of some tree of $F$, then some cycle arc is oriented backward, contradicting the observation after the definition of $\vec{\mathcal{E}}^-(F)$.

Otherwise we have a cross arc, say $v_k v_1$. We can assume that all cross arcs have endpoints in the same tree of $F$, since once we have left a component by a cross arc we can never return. Note that $v_1$ is earlier that $v_k$ because all cross arcs are directed later to earlier. Also $v_1$ is not an ancestor of $v_k$ by definition of cross arc. Assume, by induction, that $v_{i-1}$ is earlier than $v_k$ and not $v_k$'s ancestor. Now $v_{i-1}v_i$ must be either a forward or cross arc. In the former case, $v_i$ is earlier than and not an ancestor of $v_k$. In the latter case, $v_i$ must be earlier than $v_{i-1}$ and therefore earlier than $v_k$. Also, $v_i$ cannot be an ancestor of $v_k$: Since $v_{i-1}v_i$ is a cross arc, every descendant of $v_i$ is earlier than $v_{i-1}$, but by the induction hypothesis $v_k$ is later than $v_{i-1}$. Thus the induction hypothesis holds for $v_i$, which is a contradiction when $i = k$. ■

Next, we characterize the arcs in $\vec{\mathcal{E}}^-(F)$ just as we did for $\mathcal{E}(F)$. The proof is similar to that of Lemma 2.3 and is left to the reader.

**Lemma 3.2** *Suppose $G$ is a graph with spanning forest $F$ and $e \in G - F$. Then $\vec{e} \in \vec{\mathcal{E}}^-(F)$ if and only if $\vec{e}$ is of one of the following types:*

1. *$\vec{e} = uv$ is a forward arc, and $(w, v)$ is a non-inversion where $w$ is the child of $u$ on the unique $u$-$v$ path in $F$, or*

2. *$\vec{e} > \vec{f}$ where $\vec{f} \in F$ is an edge with the same endpoints and orientation as $\vec{e}$, or*

3. *$\vec{e}$ is a cross arc directed later to earlier.* ■

In our previous example the arc 15 is of type 1, arc $b = 68$ is of type 2, and all other arcs in $\vec{\mathcal{E}}^-(F)$ are of type 3.

Comparing Lemmas 2.3 and 3.2 (in particular, the conditions in the proof of the converse of the former and in the statement of the latter), we immediately obtain a corollary.

**Corollary 3.3** *Suppose $G$ is a graph with spanning forest $F$. Then*

$$|G| = |F| + |\mathcal{E}(F)| + |\vec{\mathcal{E}}^-(F)|. \blacksquare$$

We are now ready to count suborientations by initial components and number of edges. If we did not assume that $G$ was connected, the factor of $xy^{n-1}$ on the right side of the following equation would be replaced by $x^{c(G)}y^{n-c(G)}$.

**Theorem 3.4** *If $G$ has $n$ vertices, then*

$$\sum_{\mathcal{O}} x^{c(\mathcal{O})} y^{|\mathcal{O}|} = xy^{n-1}(1+y)^{\sigma^*(G)}\ t_G\left(1+x+\frac{x}{y}, \frac{1}{1+y}\right) \tag{6}$$

*where the sum is over all acyclic suborientations of $G$.*

**Proof.** Using Proposition 3.1, Corollary 3.3, and the fact that $|F| = n - c(F)$ for any spanning forest $F \subseteq G$,

$$\begin{aligned}
\sum_{\vec{\mathcal{F}}^-(\mathcal{O})=F} x^{c(\mathcal{O})} y^{|\mathcal{O}|} &= x^{c(F)} y^{|F|} \sum_{A \subseteq \vec{\mathcal{E}}^-(F)} y^{|A|} \\
&= x^{c(F)} y^{|F|} (1+y)^{|\vec{\mathcal{E}}^-(F)|} \\
&= x^{c(F)} y^{|F|} (1+y)^{|G|-|F|-|\mathcal{E}(F)|} \\
&= x^{c(F)} y^{n-c(F)} (1+y)^{|G|-n+c(F)-|\mathcal{E}(F)|} \\
&= y^n (1+y)^{|G|-n} \left(\frac{x(1+y)}{y}\right)^{c(F)} \left(\frac{1}{1+y}\right)^{|\mathcal{E}(F)|}.
\end{aligned}$$

Summing over all $F$, we obtain from equation (5),

$$\sum_{\mathcal{O}} x^{c(\mathcal{O})} y^{|\mathcal{O}|} = y^n (1+y)^{|G|-n} \frac{x(1+y)}{y}\ t_G\left(1+\frac{x(1+y)}{y}, \frac{1}{1+y}\right)$$

which agrees with (6). ■

We can rewrite this theorem using the same invariants as for subgraphs. For any digraph $D$ on $n$ vertices, let

$$\sigma(D) = c(D) - 1$$

and

$$\sigma^*(D) = |D| - n + c(D).$$

Now replace $x$ by $xy$ in equation (6)

$$\sum_{\mathcal{O}} x^{c(\mathcal{O})} y^{|\mathcal{O}|+c(\mathcal{O})} = xy^n (1+y)^{\sigma^*(G)}\ t_G\left(1+xy+x, \frac{1}{1+y}\right)$$

or

$$\sum_{\mathcal{O}} x^{\sigma(\mathcal{O})} y^{\sigma^*(\mathcal{O})} = (1+y)^{\sigma^*(G)}\ t_G\left(1+x+xy, \frac{1}{1+y}\right) \tag{7}$$

Several special cases are of interest.

To count acyclic suborientations $\mathcal{O}$ by edges without regard to the number of initial components, we set $x = 1$ in (6) and obtain

$$\sum_{\mathcal{O}} y^{|\mathcal{O}|} = y^{n-1}(1+y)^{\sigma^*(G)}\ t_G\left(2+\frac{1}{y}, \frac{1}{1+y}\right).$$

Similarly, setting $x = 1$ in (7) we get

$$\sum_{\mathcal{O}} y^{\sigma^*(\mathcal{O})} = (1+y)^{\sigma^*(G)}\ t_G\left(2+y, \frac{1}{1+y}\right).$$

In particular,

$$2^{\sigma^*(G)}\ t_G\left(3, \frac{1}{2}\right) = \text{ number of acyclic suborientations of } G.$$

On the other hand, counting such orientations by number of initial components is done by putting $y = 1$ in (6):

$$\sum_{\mathcal{O}} x^{c(\mathcal{O})} = x2^{\sigma^*(G)}\ t_G\left(1+2x, \frac{1}{2}\right).$$

To count those $\mathcal{O}$ for which $c(\mathcal{O}) = 1$, i.e., those which are initially connected, we put $x = 0$ in (7) and obtain

$$\sum_{c(\mathcal{O})=1} y^{|\mathcal{O}|} = y^{n-1}(1+y)^{\sigma^*(G)}\ t_G\left(1, \frac{1}{1+y}\right).$$

In particular,

$$2^{\sigma^*(G)}\ t_G\left(1, \frac{1}{2}\right) = \text{ number of initially connected acyclic suborientations of } G.$$

Finally, to count acyclic orientations of $G$, i.e., those $\mathcal{O}$ with $|\mathcal{O}| = |G|$, we divide (6) by $y^{|G|}$ and take $y \to \infty$. The result is

$$\sum_{|\mathcal{O}|=|G|} x^{c(\mathcal{O})} = x\ t_G(1+x, 0). \tag{8}$$

In particular,

$$\begin{aligned} t_G(2,0) &= \text{ number of acyclic orientations of } G\text{, and} \\ t_G(1,0) &= \text{ number of initially connected acyclic orientations of } G. \end{aligned}$$

This interpretation of $t_G(2,0)$ was found by Stanley in [38], while Greene and Zaslavsky [21] discovered the one for $t_G(1,0)$. These authors expressed their results in terms of chromatic polynomials.

# 4 Subdigraphs

Let $G$ be a graph. A *directed subgraph* or *subdigraph* of $G$ is a digraph $D$ that contains up to one copy of each orientation of every edge of $G$. Thus we permit both orientations of an edge (including loops) to appear in a subdigraph, as opposed to a suborientation where only one is permitted.

We now consider greatest-neighbor DFS on the set of all subdigraphs of $G$. The only difference from the subgraph case is that now we are constrained to follow the directions on the arcs. If digraph $D$ has greatest-neighbor forest F, we write $\vec{\mathcal{F}}^+(D) = F$. There is also the set $\vec{\mathcal{E}}^+(F)$ of *(directed) greatest-neighbor externally active orientations* $\vec{e}$ of edges of $G$ such that $\vec{\mathcal{F}}^+(F \uplus \vec{e}\,) = F$. Notice that because of the disjoint union, we have $\vec{e} \notin F$. However, if $uv \in F$ then we always have $vu \in \vec{\mathcal{E}}^+(F)$.

The next four results are similar to those we have seen in the previous sections, so we will only indicate a proof of the third. In what follows, if $F$ is a forest then an *$\mathcal{E}$-active edge* is an edge $e$ in $\mathcal{E}(F)$, i.e., $e$ is greatest-neighbor externally active in the undirected sense. All other edges of $G - F$ will be called *$\mathcal{E}$-passive.*

**Proposition 4.1** *If $D$ is any subdigraph and $F$ is any spanning forest of $G$ then*

$$\vec{\mathcal{F}}^+(D) = F \quad \iff \quad F \subseteq D \subseteq F \uplus \vec{\mathcal{E}}^+(F). \blacksquare$$

**Lemma 4.2** *Suppose $G$ is a graph with spanning forest $F$. Then $\vec{e} \in \vec{\mathcal{E}}^+(F)$ if and only if $\vec{e}$ is of one of the following types:*

1. *$e$ is an $\mathcal{E}$-active arc directed forward.*

2. *$e$ is any arc of $G$ directed later to earlier.* $\blacksquare$

**Corollary 4.3** *Suppose $G$ is a graph with spanning forest $F$. Then*

$$|G| = |\vec{\mathcal{E}}^+(F)| - |\mathcal{E}(F)|.$$

**Proof.** The edges of $G$ can be partitioned into those in $F$, those that are $\mathcal{E}$-active and those that are $\mathcal{E}$-passive. The following table lists the number of times each sort of edge is counted in $\vec{\mathcal{E}}^+(F)$ and $\mathcal{E}(F)$.

| edges | $\vec{\mathcal{E}}^+(F)$ | $\mathcal{E}(F)$ |
|---|---|---|
| $F$ | 1 (backward) | 0 |
| $\mathcal{E}$-active | 2 (forward and backward) | 1 |
| $\mathcal{E}$-passive | 1 (later to earlier) | 0 |

Since the net difference is 1 in each case, the result follows. ■

**Theorem 4.4** *If $G$ has $n$ vertices, then*

$$\sum_D x^{c(D)} y^{|D|} = x y^{n-1} (1+y)^{|G|}\, t_G\left(1 + \frac{x}{y}, 1+y\right) \tag{9}$$

*and*

$$\sum_D x^{\sigma(D)} y^{\sigma^*(D)} = (1+y)^{|G|}\, t_G\,(1+x, 1+y) \tag{10}$$

*where the sum is over all subdigraphs of $G$.* ■

As special cases, we can count subdigraphs by edges or $\sigma^*$ invariant:

$$\begin{aligned}
\sum_D y^{|D|} &= y^{n-1}(1+y)^{|G|}\, t_G\left(1+\frac{1}{y}, 1+y\right) \;=\; (1+y)^{2|G|} \\
\sum_D y^{\sigma^*(D)} &= (1+y)^{|G|}\, t_G\,(2, 1+y) \\
2^{|G|}\, t_G(2,2) &= \text{number of subdigraphs of } G \;= 4^{|G|},
\end{aligned}$$

or by number of initial components:

$$\begin{aligned}
\sum_D x^{c(D)} &= x 2^{|G|}\, t_G\,(1+x, 2) \\
\sum_{c(D)=1} y^{|D|} &= y^{n-1}(1+y)^{|G|}\, t_G\,(1, 1+y) \\
2^{|G|}\, t_G(1,2) &= \text{number of initially connected subdigraphs of } G.
\end{aligned}$$

From equations (4) and (10), we see that

$$\sum_D x^{\sigma(D)} y^{\sigma^*(D)} = (1+y)^{|G|} \sum_{H \subseteq G} x^{\sigma(H)} y^{\sigma^*(H)}.$$

This equality can also be proved directly by exhibiting a $2^{|G|}$-to-1 map from subdigraphs $D$ of $G$ to subgraphs $H \subseteq G$ that preserves the appropriate invariants as follows: From Lemma 4.2, $D$ can be represented by a triple $(F, E, A)$ where $F = \vec{\mathcal{F}}^+(D)$, $E$ is the set of directed $\mathcal{E}$-active edges in $D$, and $A$ is the rest of the arcs of $D$ (so the corresponding edges could be an arbitrary subset of $G$). Similarly, Lemma 2.3 shows that $H$ can be represented by a pair $(F, E)$ where $F = \mathcal{F}^+(H)$ and $E$ is the set of $\mathcal{E}$-active edges in $H$. It is easy to verify that the projection map

$$(F, E, A) \to (F, E)$$

(where we change arcs to edges in the image) has the desired properties.

# 5 Complete graphs

We will now show how our results on orientations and subdigraphs can be combined with the generating function for the Tutte polynomial of a complete graph to obtain various new generating functions, some of which generalize results already in the literature. For brevity, let $t_n(x,y) = t_{K_n}(x,y)$. Tutte [41, equation (17)] obtained an equation equivalent to the following which can be derived using the exponential formula.

**Theorem 5.1** *The Tutte polynomial of the complete graph has exponential generating function*

$$\sum_{n\geq 1} t_n(x,y)\frac{u^n}{n!} = \frac{1}{x-1}\left\{\left[\sum_{n\geq 0} y^{\binom{n}{2}}(y-1)^{-n}\frac{u^n}{n!}\right]^{(x-1)(y-1)} - 1\right\}. \blacksquare$$

Next we find the generating function for acyclic digraphs, which are just acyclic suborientations $\mathcal{O} \subseteq K_n$. To do this, it will be convenient to define the *graphic generating function* of a sequence $(a_n)_{n\geq 0}$ to be

$$\sum_{n\geq 0} a_n \frac{u^n}{(1+y)^{\binom{n}{2}} n!}.$$

According to equation (6), the count of acyclic digraphs on $n$ vertices by number of arcs and initial components is given by

$$a_n(x,y) \stackrel{\text{def}}{=} \sum_{\mathcal{O}\subseteq K_n} x^{c(\mathcal{O})} y^{|\mathcal{O}|} = xy^{n-1}(1+y)^{\binom{n-1}{2}}\; t_n\left(1+x+\frac{x}{y}, \frac{1}{1+y}\right).$$

Applying the previous theorem yields

$$\begin{aligned}
\sum_{n\geq 1} \frac{a_n(x,y)}{y^n(1+y)^{\binom{n-1}{2}}}\frac{u^n}{n!} &= \frac{x}{y}\,\frac{1}{x+\frac{x}{y}}\left\{\left[\sum_{n\geq 0}\frac{1}{(1+y)^{\binom{n}{2}}}\left(\frac{-y}{1+y}\right)^{-n}\frac{u^n}{n!}\right]^{(x+\frac{x}{y})(\frac{-y}{1+y})} - 1\right\}\\
&= \frac{1}{1+y}\left\{\left[\sum_{n\geq 0}(-1)^n\frac{(1+y)^n}{y^n}\frac{u^n}{(1+y)^{\binom{n}{2}} n!}\right]^{-x} - 1\right\}. \qquad (11)
\end{aligned}$$

If we define $a_0(x,y) = 1$ and replace $u$ by $\frac{yu}{1+y}$, then this last result simplifies.

**Corollary 5.2** *The graphic generating function for acyclic digraphs is*

$$\sum_{n\geq 0} a_n(x,y)\frac{u^n}{(1+y)^{\binom{n}{2}} n!} = \left[\sum_{n\geq 0}(-1)^n\frac{u^n}{(1+y)^{\binom{n}{2}} n!}\right]^{-x}. \blacksquare$$

Stanley [38] and Robinson [35] obtained this result when $x = y = 1$, as did Liskovets [29] and Rodionov [36] when $x = 1$. When $x$ and $y$ are integers with $y \geq 0$, Stanley also gives an interpretation to these graphic generating functions in terms of the theory of posets of full binomial type developed by himself, Doubilet and Rota [11].

We can derive the generating function for initially connected acyclic digraphs counted by number of arcs using

$$c_n(y) \stackrel{\text{def}}{=} \left.\frac{a_n(x,y)}{x}\right|_{x=0}.$$

Dividing equation (11) by $x$ and letting $x \to 0$ yields the desired formula.

**Corollary 5.3** *The graphic generating function for initially connected acyclic digraphs is*

$$\sum_{n\geq 1} c_n(y) \frac{u^n}{(1+y)^{\binom{n}{2}} n!} = \ln \left[ \sum_{n\geq 0} (-1)^n \frac{u^n}{(1+y)^{\binom{n}{2}} n!} \right]^{-1}. \blacksquare$$

Putting together these last two theorems, we see that

$$\sum_{n\geq 0} a_n(x,y) \frac{u^n}{(1+y)^{\binom{n}{2}} n!} = \exp \left[ x \sum_{n\geq 1} c_n(y) \frac{u^n}{(1+y)^{\binom{n}{2}} n!} \right]$$

This is a special case of a more general exponential formula for digraphs which does not seem to have been stated before.

**Theorem 5.4 (Exponential formula for digraphs)** *Let $\mathcal{D}$ be a class of initially connected labeled digraphs with the property that an order-preserving change of labels does not affect membership in $\mathcal{D}$. Let $c_n(y)$ count digraphs in $\mathcal{D}$ on the label set $\{1, 2, \ldots, n\}$ by number of arcs. Then*

$$\exp \left[ x \sum_{n\geq 1} c_n(y) \frac{u^n}{(1+y)^{\binom{n}{2}} n!} \right] = \sum_{k,m,n\geq 0} b_{k,m,n}\, x^k y^m \frac{u^n}{(1+y)^{\binom{n}{2}} n!} \tag{12}$$

*where $b_{k,m,n}$ is the number of digraphs on $\{1, 2, \ldots, n\}$ with $m$ arcs, $k$ initial components, and every such component in $\mathcal{D}$.* $\blacksquare$

**Proof.** Taking the coefficient of $x^k u^n$ on both sides of equation (12), it suffices to show

$$\frac{1}{(1+y)^{\binom{n}{2}} n!} \sum_{m\geq 0} b_{k,m,n}\, y^m = \frac{1}{k!} \sum_{n_1+\cdots+n_k=n} \prod_{i=1}^{k} \frac{c_{n_i}(y)}{(1+y)^{\binom{n_i}{2}} n_i!}$$

where the sum is over all ordered partitions of $n$. Equivalently

$$\sum_{m\geq 0} b_{k,m,n}\, y^m = \frac{1}{k!}\sum_{n_1+\cdots+n_k=n}\binom{n}{n_1,\ldots,n_k} c_{n_1}(y)\cdots c_{n_k}(y)\,(1+y)^{\sum_{i<j} n_i n_j}$$

The left side of this formula just counts digraphs $D$ on $n$ vertices with $k$ initial components by number of arcs. But the right sums to the same thing. The multinomial coefficient counts the number of ways to partition the vertices of $D$ into $k$ ordered subsets for the initial components. Summing over all ordered partitions of $n$ and then dividing by $k!$ gives coefficients which count unordered partitions of the vertices. The $c_{n_i}(y)$ give the arc count for each component. And the power of $1+y$ accounts for the arcs between components which must all be directed from a later to an earlier component. ■

A general theory of exponential formulas has been developed by Stanley [39], but this result does not seem to be a consequence. In the example we have been considering, the set $\mathcal{D}$ consists of all initially connected acyclic digraphs.

As a further demonstration, we can count digraphs without the acyclicity condition. Let $d_n(x,y) = \sum_D x^{c(D)} y^{|D|}$ (respectively, $e_n(y) = \sum_D y^{|D|}$) where the sum is over all digraphs (respectively, all initially connected digraphs) on $n$ vertices. The graphic generating function for all digraphs by number of arcs is

$$\sum_{n\geq 0}(1+y)^{2\binom{n}{2}}\frac{u^n}{(1+y)^{\binom{n}{2}} n!} = \sum_{n\geq 0}(1+y)^{\binom{n}{2}}\frac{u^n}{n!}$$

Using our Exponential Formula, we immediately get the following result.

**Corollary 5.5** *The graphic generating function for $d_n(x,y)$ is*

$$\sum_{n\geq 0} d_n(x,y)\frac{u^n}{(1+y)^{\binom{n}{2}} n!} = \left[\sum_{n\geq 0}(1+y)^{\binom{n}{2}}\frac{u^n}{n!}\right]^x .$$

*The graphic generating function for $e_n(y)$ is*

$$\sum_{n\geq 1} e_n(y)\frac{u^n}{(1+y)^{\binom{n}{2}} n!} = \ln\left[\sum_{n\geq 0}(1+y)^{\binom{n}{2}}\frac{u^n}{n!}\right]. \blacksquare$$

# 6 Neighbors-first search

A specialization of $t_n(x,y)$ that has received some attention is $t_n(y) \stackrel{\text{def}}{=} t_n(1,y)$. Mallows and Riordan [30] first studied this polynomial as the inversion enumerator for trees. See also the book of Foata [12, pp. 144–147] and the papers of

Gessel, Sagan and Yeh [19] and Gessel [17]. Kreweras [28] has given a number of other interpretations to this polynomial which have been further studied by Moszkowski [31]. See also Beissinger [3], who gives a bijective proof that $t_n(1, y)$ counts inversions. In Section 9 we will give a table of these polynomials.

For completeness, we state the connection with the inversion polynomial. It follows directly from Lemma 2.3 applied to $G = K_n$.

**Proposition 6.1** *If $T$ is a tree, let $\operatorname{inv} T$ stand for the number of inversions of $T$. Then*

$$t_n(y) = \sum_T y^{\operatorname{inv} T}$$

*where the sum is over all trees with $n$ vertices.* ■

We will now give another interpretation of $t_n(y)$ using a modified version of DFS which is sometimes called *neighbors-first search* or NFS (see [7, p. 154]). The following steps are applied to a graph $H$ to build an NFS forest $F$. Note that marking and searching a vertex are now two separate actions.

NFS1 Let $F = \emptyset$.

NFS2 Let $v$ be the least unmarked vertex in $V$ and mark $v$.

NFS3 Search $v$ by marking all neighbors of $v$ that have not been marked and adding to $F$ all edges from $v$ to theses vertices.

NFS4 Recursively search all the vertices marked in NSF3 in increasing order, stopping when every vertex that has been marked has also been searched.

NSF5 If there are unmarked vertices, then return to NSF2. Otherwise, stop.

Thus NFS searches nodes in a depth-first manner but marks children in a locally breadth-first manner. In choosing the vertex $u$ in NFS2, we will always pick the one with smallest label and use the smallest ordered edge. Denote the resulting forest by $F = \mathcal{F}_N(H)$. As an example, for the graph $H$ in Figure 5 we start at vertex 1, designating 3 and 4 as its children. Next we search node 3 and mark 5, 6 and 8 as its offspring. Note that 4 cannot be a child of 3 since it is already a child of 1. The search now continues at 5, and so forth.

Observe that traversing a forest $F$ by NFS gives a linear ordering to the children of each vertex, i.e., the order in which we search them from smallest to largest label. We will display this as a left-to-right order of the siblings when we draw $F$ in the plane and use corresponding terminology.

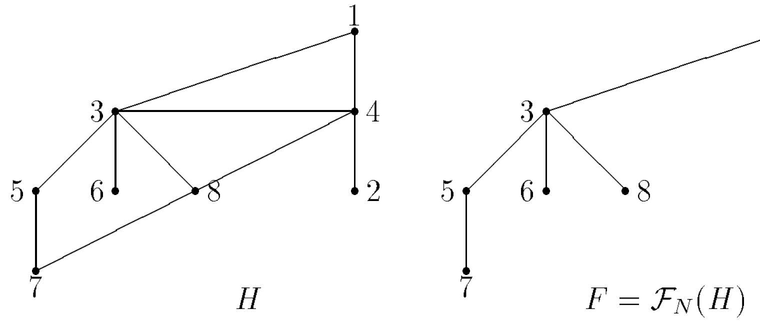


Figure 5: A graph $H$ and NSF forest $F$

As usual, given a spanning forest $F$ of a graph $G$, we define $\mathcal{E}_N(F)$, the set of edges *externally active with respect to NFS*, to be those edges $e$ in $G - F$ such that

$$\mathcal{F}_N(F \cup e) = F.$$

The next set of results should be easy for the reader to prove by mimicking what we did in Section 2. Proofs are therefore omitted.

**Proposition 6.2** *If $H$ is any subgraph and $F$ is any spanning forest of $G$ then*

$$\mathcal{F}_N(H) = F \quad \Longleftrightarrow \quad F \subseteq H \subseteq F \uplus \mathcal{E}_N(F). \blacksquare$$

**Proposition 6.3** *If $G$ is a connected graph, then*

$$t_G(1 + x, y) = \sum_{F \subseteq G} x^{\sigma(F)} y^{|\mathcal{E}_N(F)|}$$

*where the sum is over all spanning forests of $G$. In particular*

$$t_n(y) = \sum_T y^{|\mathcal{E}_N(T)|} \tag{13}$$

*where the sum is over all trees on $n$ vertices.* $\blacksquare$

**Theorem 6.4** *Suppose $G$ is a graph with spanning forest $F$ and $e \in G - F$. Then $e \in \mathcal{E}_N(F)$ if and only if $e$ is of one of the following types:*

1. *$e = \{u, v\}$ where $v$ is a descendant of $u$'s parent, and $w < u$ where $w$ is the sibling of $u$ on the unique path from their parent to $v$ in $F$, or*

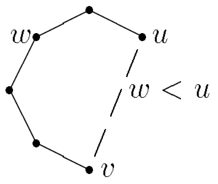


Figure 6: The first case of Theorem 6.4

*2. $e > f$ where $f \in F$ is an edge with the same endpoints as $e$, or*

*3. $e$ is a loop.* ■

Since our applications will all be to $G = K_n$, only the first of these three cases really matters. A schematic diagram of this case is given in Figure 6.

It follows from Propositions 6.1 and 6.3 that the distribution of $|\mathcal{E}_N(T)|$ for labeled NFT trees is the same as that for $\operatorname{inv} T$. We digress briefly to note that the the distribution of external activities for unlabeled ordered trees is given by the $q$-Catalan numbers studied by Andrews [1], Fürlinger and Hofbauer [14], and Krattenthaler [27]. Any unlabeled ordered tree can be given an *NFT labeling* by labeling the root as 1 and then making sure that the labels on the children of every vertex increase from left to right. Thus we can let the external activity of an unlabeled tree $T$ be the external activity of any NFT labeling of $T$ as a spanning tree of a complete graph. Now define polynomials $C_n(q)$ by $C_0(q) = 1$ and

$$C_n(q) = \sum_{k=0}^{n-1} q^k C_k(q) C_{n-k-1}(q). \tag{14}$$

The first few values are

$$C_0(q) = C_1(q) = 1, \qquad C_2(q) = 1 + q, \qquad C_3(q) = 1 + 2q + q^2 + q^3,$$
$$C_4(q) = 1 + 3q + 3q^2 + 3q^3 + 2q^4 + q^5 + q^6.$$

If we compute the external activities of the unlabeled ordered trees on 3 edges (see Figure 7), then we obtain

$$\sum_{|T|=3} q^{|\mathcal{E}_N(T)|} = 1 + 2q + q^2 + q^3.$$

This is evidence for the next theorem.

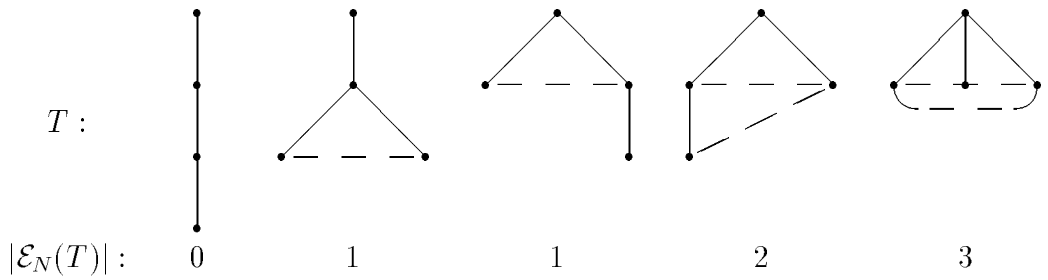


Figure 7: Externally active edges for unlabeled ordered trees on 3 edges

**Theorem 6.5** *We have*

$$C_n(q) = \sum_{|T|=n} q^{|\mathcal{E}_N(T)|}$$

*where the sum is over all unlabeled ordered trees $T$ with $n$ edges.*

**Proof.** It suffices to show that the tree sum satisfies the recursion (14). Now any tree $T$ can be decomposed into two trees $T'$ and $T''$ where

$$\begin{aligned} T' &= \text{rightmost subtree of the root } r, \text{ and} \\ T'' &= T - T' \end{aligned}$$

Here we make the convention that the edge joining $r$ to $T'$ is removed in $T - T'$. Also, if $r$ has only one subtree, it is considered rightmost so that $T'' = r$ in this case. But if $|T| = n$ and $|T''| = k$, then $|T'| = n - k - 1$. So by case 1 of Theorem 6.4

$$q^{|\mathcal{E}_N(T)|} = q^k q^{|\mathcal{E}_N(T')|} q^{|\mathcal{E}_N(T'')|}.$$

Summing over all $T$, we obtain the desired result by induction. ■

We now return to the main stream of our development. For the rest of this section and the next, all of our external activities will be with respect to the graph $K_n$. In this case, given any tree $T$, we can derive a simple formula for $|\mathcal{E}_N(T)|$. Let $v_1, v_2, \ldots, v_n$ be the order in which the nodes of $T$ are first searched using NFS. Note that this is a depth-first order. Define the *prefix code* of $T$ to be the sequence

$$c(T) = c_1, c_2, \ldots, c_n$$

where $c_i$ is the number of children of vertex $v_i$. We could also define $c(T)$ recursively by

$$c(T) = c_1, c(T_1), c(T_2), \ldots, c(T_k)$$

where $T_1, T_2, \ldots, T_k$ are the subtrees of $v_1$ listed in order of increasing labels of their roots. For example, the nodes of the tree in Figure 5 are searched in the order

$$v_1, v_2, \ldots, v_8 = 1, 3, 5, 7, 6, 8, 4, 2 \tag{15}$$

which gives it prefix code

$$c_1, c_2, \ldots, c_8 = 2, 3, 1, 0, 0, 0, 1, 0.$$

**Theorem 6.6** *If $T$ is a tree with prefix code $c_1, c_2, \ldots, c_n$ then*

$$\begin{aligned} |\mathcal{E}_N(T)| &= (c_1 - 1) + (c_1 + c_2 - 2) + \cdots + (c_1 + c_2 + \cdots + c_{n-1} - n + 1) \\ &= (n-1)c_1 + (n-2)c_2 + \cdots + c_{n-1} - \binom{n}{2} \end{aligned}$$

Using our previous example

$$\begin{aligned} \sum_{k=1}^{n-1} (n-k)c_k - \binom{n}{2} &= 7 \cdot 2 + 6 \cdot 3 + 5 \cdot 1 + 4 \cdot 0 + 3 \cdot 0 + 2 \cdot 0 + 1 \cdot 1 - \binom{8}{2} \\ &= 10 \end{aligned}$$

while

$$\mathcal{E}_N(T) = \{\{4,3\}\ \{4,5\}\ \{4,6\}\ \{4,7\}\ \{4,8\}\ \{6,5\}\ \{6,7\}\ \{8,5\}\ \{8,6\}\ \{8,7\}\}$$

which has 10 elements.

**Proof of Theorem 6.6.** It suffices to show that the term $c_1 + c_2 + \cdots + c_i - i$ counts all externally active edges whose left end is $v_{i+1}$. We will do this by induction on $i$. This is clearly true for $i = 0$. For $i > 0$, we distinguish two cases.

If $c_i > 0$, then $v_{i+1}$ is the leftmost child of $v_i$. Now $\{v_i, u\}$ active implies that so is $\{v_{i+1}, u\}$ (Theorem 6.4), yielding $c_1 + c_2 + \cdots + c_{i-1} - i + 1$ edges. Also, there are active edges from $v_{i+1}$ to each of its siblings, for $c_i - 1$ more edges. These are the only active edges and the total is correct.

If $c_i = 0$, then $v_i$ is a leaf and we get to $v_{i+1}$ by backtracking. But then $v_{i+1}$ was the leftmost of all the vertices joined to $v_i$ by externally active edges. So $v_{i+1}$ has exactly one less $(= c_i - 1)$ active edge than $v_i$ did. Thus we are finished by induction. ∎

We can use the NFS interpretation of $t_n(y)$ to give a combinatorial proof of an identity for its generating function first proved by other means in [16].

**Theorem 6.7** *Let*

$$J(u) = \sum_{n \geq 0} t_{n+1}(y) \frac{u^n}{n!}$$

*then*

$$J(u) = \sum_{n \geq 0} y^{\binom{n}{2}} J(u) J(yu) \cdots J(y^{n-1}u) \frac{u^n}{n!}$$

**Proof.** Taking the coefficient of $\frac{u^n}{n!}$ on both sides, we get the equivalent statement

$$\begin{aligned} t_{n+1}(y) &= \sum_{\substack{k \geq 0 \\ n_1 + \cdots + n_k + k = n}} y^{\binom{k}{2}} \left[ y^{(k-1)n_1} t_{n_1+1}(y) \right] \left[ y^{(k-2)n_2} t_{n_2+1}(y) \right] \cdots \frac{n!}{n_1! n_2! \cdots n_k! k!} \\ &= \sum_{\substack{k \geq 0 \\ n_1 + \cdots + n_k + k = n}} \binom{n}{k, n_1, n_2, \ldots, n_k} t_{n_1+1}(y) t_{n_2+1}(y) \cdots \ y^{\sum_{i=1}^{k} (k-i)(n_i+1)}. \end{aligned}$$

where the factor involving $t_{n_i+1}(y)$ comes from $J(y^{k-i}u)$.

To see that this last expression enumerates trees $T$ by externally active edges, consider the subtrees $T_1, T_2, \ldots, T_k$ of the root of $T$. Suppose these trees have roots $w_1, w_2, \ldots, w_k$ and $n_1, n_2, \ldots, n_k$ other vertices respectively. Then the multinomial coefficient counts the number of ways to pick the roots and then the other sets of vertices.

The active edges $\{v, w\}$ are of two types:

- edges where $v$ and $w$ are in the same $T_i$, and
- edges where $v \in T_i$ and $w = w_j$ for some $i < j$

Edges of the first sort are accounted for by $t_{n_1+1}(y) t_{n_2+1}(y) \cdots t_{n_k+1}(y)$ while those of the second are taken care of by the power of $y$. ∎

We end with a characterization of forests in terms of prefix codes that will help us in Section 7. Since it is well known we omit the proof.

**Theorem 6.8** *The sequence $c_1, c_2, \ldots, c_n$ is a prefix code for a tree if and only if*

1. $\displaystyle\sum_{i=1}^{j} (c_i - 1) \geq 0 \qquad$ *for $j < n$, and*
2. $\displaystyle\sum_{i=1}^{n} (c_i - 1) = -1$ ∎

Notice that the preceding conditions could be rewritten as

$$\begin{aligned} \sum_{i=1}^{j} c_i &\geq j \qquad \text{for } j < n, \text{ and} \\ \sum_{i=1}^{n} c_i &= n-1. \end{aligned}$$

We can make the characterization in Theorem 6.8 even stronger by using parent functions. Suppose we are given a tree $T$ having vertices $\{1, \ldots, n\}$ and NFS order $v_1 = 1, v_2, \ldots, v_n$. The corresponding *parent function* is $p : \{2, \ldots, n\} \to \{1, \ldots, n\}$ defined by $p(i) = j$ if the vertex labeled $i$ has parent $v_j$. Returning to the tree in Figure 5 with NFS order given by (15), we see that it has parent function

$$p(2) = 7,\ p(3) = 1,\ p(4) = 1,\ p(5) = 2,\ p(6) = 2,\ p(7) = 3,\ p(8) = 2.$$

Observe that $|p^{-1}(j)| = c_j$. This should motivate the following result whose proof, since it follows easily from the previous theorem, is omitted.

**Corollary 6.9** *The function $p : \{2, \ldots, n\} \to \{1, \ldots, n\}$ is a parent function for a tree if and only if*

$$|p^{-1}(1) \cup p^{-1}(2) \cup \cdots \cup p^{-1}(j)| \geq j \qquad \textit{for } j < n$$

# 7 Hashing and parking functions

We now describe an application of $t_n(y)$ to hash coding, which is a method for storing and retrieving data efficiently. Knuth [24, Chapter 6] gives a comprehensive account of storage and retrieval methods, and in particular of hash coding (Section 6.4). The hashing technique we discuss here is called "open addressing with linear probing." It was analyzed earlier by Knuth in [25]. We consider only the storage aspect; retrieval is similar and is discussed in detail by Knuth.

Suppose we have $m$ boxes labeled 1 to $m$, and $n < m$ objects labeled 1 to $n$ which are to be put into the boxes. Each object $i$ has a preferred box $h(i)$, where the function $h$ is called a *hash function*. We now insert the objects in the order 1, 2, ..., $n$ into the boxes. When we insert object $i$, we place it into box $h(i)$ if this box is empty. Otherwise, we *probe* boxes $h(i) + 1, h(i) + 2, \ldots$ in turn and place object $i$ into the first empty box we find. Box numbers are taken modulo $m$, so that box 1 is probed after box $m$. Since there are more boxes than objects, every object will eventually be placed into a box. Given a hash function $h$, which may be an arbitrary function from $\{1, 2, \ldots, n\}$ to $\{1, 2, \ldots, m\}$, we let $B(h)$ be the

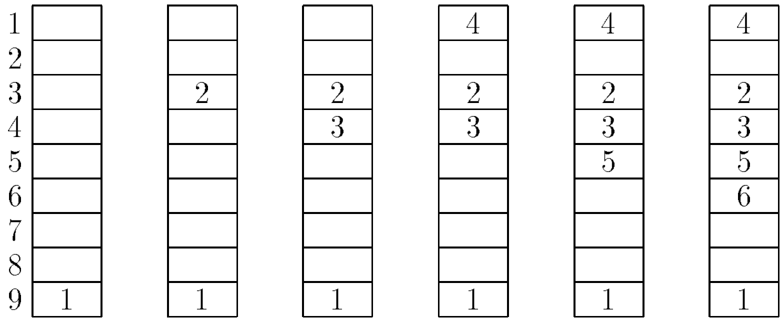

Figure 8: Open address hashing

number of times an occupied box is probed during the insertion process. (Note that Knuth counts as a probe the box into which an object is inserted, but we do not.) By way of illustration, consider the array of $m = 9$ boxes in Figure 8 where the box numbers are given on the far left. Reading the diagram from left to right shows the placement of $n = 6$ objects using the hash function

$$h(1) = 9,\ h(2) = 3,\ h(3) = 3,\ h(4) = 9,\ h(5) = 4,\ h(6) = 3.$$

The number of probes is

$$B(h) = 0 + 0 + 1 + 1 + 1 + 3 = 6.$$

The following lemma, which appears in [24, pp. 530–531], is very useful.

**Lemma 7.1 (Rearrangement Lemma)** *Suppose $h$ and $g$ are two hash functions such that the sequence $g(1), \ldots, g(n)$ is a rearrangement of $h(1), \ldots, h(n)$. Then*

1. *the same boxes are filled in the insertion process for $h$ and $g$, and*
2. *$B(h) = B(g)$.* ■

We now study the distribution of $B(h)$ among the $m^n$ possible hash functions $h : \{1, 2, \ldots, n\} \to \{1, 2, \ldots, m\}$, where $n < m$. Let $K_{n,m,i}$ be the number of such $h$ with $B(h) = i$. Also, let $L_{n,m,i}$ be the number of these functions with the property that after all $n$ objects are inserted, box $m$ is empty. Since all boxes are equally likely to be empty, we have

$$L_{n,m,i} = \frac{k}{m} K_{n,m,i}, \tag{16}$$

where $k = m - n$ is the number of empty boxes.

We now examine the polynomial

$$L_{n,m}(y) = \sum_{i \geq 0} L_{n,m,i}\ y^i. \tag{17}$$

Suppose we perform the insertions corresponding to a hash function counted by $L_{n,m}(y)$. Consider the sequence of boxes after completing these insertions. This sequence can be broken up into $k = m - n$ subsequences, each of which consists of zero or more filled boxes followed by an empty box. Since no object will ever probe any of the $k$ empty boxes, the sequence of boxes can be obtained by decomposing the hash function into $k$ functions and doing the insertions for each separately. In the example from Figure 8, there are $k = 3$ subsequences, consisting of boxes $\{3, 4, 5, 6, 7\}$, $\{8\}$, and $\{9, 1, 2\}$. By the Rearrangement Lemma and the properties of exponential generating functions, we have

$$\sum_{n \geq 0} L_{n,n+k}(y) \frac{u^n}{n!} = \left[ \sum_{n \geq 0} L_{n,n+1}(y) \frac{u^n}{n!} \right]^k. \tag{18}$$

It remains to determine $L_{n,n+1}(y)$.

The functions counted by $L_{n,n+1}(y)$ are called *parking functions*: they are hash functions $p : \{1, \ldots, n\} \to \{1, \ldots, n+1\}$ that leave box $n+1$ empty. The name derives from the scenario [24, p. 545, exercise 29] in which the boxes are interpreted as parking spaces and the objects are cars trying to park, with the hash function giving the preferred spot of each car. The term "parking function" was coined by Konheim and Weiss [26]. We have the following characterization of parking functions.

**Theorem 7.2** *The hash function $p : \{1, \ldots, n\} \to \{1, \ldots, n+1\}$ is a parking function if and only if*

$$|p^{-1}(1) \cup p^{-1}(2) \cup \cdots \cup p^{-1}(j)| \geq j \text{ for } j < n+1.$$

*Furthermore, in this case*

$$B(p) = nc_1 + (n-1)c_2 + \cdots + c_n - \binom{n+1}{2},$$

*where $c_i = |p^{-1}(i)|$.*

**Proof.** Suppose that $p$ is a parking function. Since the first $j$ boxes can be filled only from objects in $p^{-1}(1) \cup p^{-1}(2) \cup \cdots \cup p^{-1}(j)$, we must have $|p^{-1}(1) \cup p^{-1}(2) \cup$

$\cdots \cup p^{-1}(j)| \geq j$ for $j < n+1$. The converse follows from the observation that if $|p^{-1}(1) \cup p^{-1}(2) \cup \cdots \cup p^{-1}(j)| \geq j$ then box $j$ will be filled.

To prove the formula for $B(p)$ it suffices to show that $c_1 + c_2 + \ldots + c_j - j$ counts the number of times box $j$ is probed after it is filled, for $j = 1, 2, \ldots, n$. But $c_1 + c_2 + \cdots + c_j$ is the total number of objects that start their search in box $j$ or before. And of these, the first $j$ objects will occupy the first $j$ boxes, leaving $c_1 + c_2 + \ldots + c_j - j$ to probe box $j$. ∎

Comparison of Theorem 7.2 with Theorem 6.6 and Corollary 6.9 shows that there is a bijection between NFS trees $T$ on $\{1, 2, \ldots, n+1\}$ and parking functions $p : \{1, \ldots, n\} \to \{1, \ldots, n+1\}$ such that $\mathcal{E}_N(T) = B(p)$. Thus

$$L_{n,n+1}(y) = t_{n+1}(y). \tag{19}$$

This was first proved by Kreweras [28], who studied the functions satisfying the property of Theorem 7.2, but did not identify them as parking functions. For further work on parking functions, see Schützenberger [37], Riordan [34], Foata and Riordan [13], and Moszkowski [31].

In analyzing the performance of hash coding as a storage method, one wants to know the expected value of $B(h)$ over all hash functions $h : \{1, 2, \ldots, n\} \to \{1, 2, \ldots, m\}$, assuming that all are equally likely. Although Knuth computes this expected value without knowing $L_{n,m}(y)$, it is interesting to see how this value can be derived from our results.

The expected value of $B(h)$ over all hash functions is clearly the same as the expected value of $B(h)$ over hash functions that leave box $m$ empty, which is $L'_{n,m}(1)/L_{n,m}(1)$. By (16) and (17),

$$L_{n,n+k}(1) = \frac{k}{n+k}(n+k)^n = k(n+k)^{n-1}.$$

Also, by (18) and (19) we have

$$\sum_{n\geq 0} L'_{n,n+k}(1)\frac{u^n}{n!} = k\left[\sum_{n\geq 0} t_{n+1}(1)\frac{u^n}{n!}\right]^{k-1} \sum_{n\geq 0} t'_{n+1}(1)\frac{u^n}{n!}. \tag{20}$$

We know that $t_{n+1}(1) = (n+1)^{n-1}$. It remains to evaluate $t'_{n+1}(1)$. By equation (4), we have

$$t_{n+1}(y) = \sum_H (y-1)^{\sigma^*(H)}$$

where the sum is over all connected graphs on $\{1, 2, \ldots, n+1\}$. Then

$$t'_{n+1}(1) = \sum_H \sigma^*(H)\ (y-1)^{\sigma^*(H)-1}\Big|_{y=1}.$$

The only non-zero terms in this sum occur when $1 = \sigma^*(H) = |H| - (n+1) + c(H)$. Since $H$ is connected, this implies it must be unicyclic. The number of such graphs is known [32, 43]. Substituting this value into the previous equation gives

$$t'_{n+1}(1) = \frac{1}{2}\sum_{j=3}^{n+1}\binom{n+1}{j} j!\,(n+1)^{n-j}. \tag{21}$$

Now let

$$T = T(u) = \sum_{n\geq 0}(n+1)^{n-1}\frac{u^n}{n!}. \tag{22}$$

It is well known that

$$\frac{T^j}{1-uT} = \sum_{l\geq 0}(l+j)^l\frac{u^l}{l!}.$$

See, for example, Riordan's book [33, p. 147]. It follows that

$$\sum_{n=j-1}^{\infty}\binom{n+1}{j} j!\,(n+1)^{n-j}\frac{u^n}{n!} = \frac{u^{j-1}T^j}{1-uT}.$$

Combining this equation with (20), (21), and (22) yields

$$\begin{aligned}
\sum_{n\geq 0} L'_{n,n+k}(1)\frac{u^n}{n!} &= \frac{k}{2}T^{k-1}\sum_{j=3}^{\infty}\frac{u^{j-1}T^j}{1-uT} \\
&= \frac{k}{2}\sum_{i=2}^{\infty}\frac{u^iT^{i+k}}{1-uT} \\
&= \frac{k}{2}\sum_{i=2}^{\infty}\sum_{l\geq 0}(l+i+k)^l\frac{u^{l+i}}{l!} \\
&= \sum_{n\geq 0}\frac{u^n}{n!}\sum_{i=2}^{n}\frac{1}{2}\binom{n}{i} i!\,k(n+k)^{n-i}.
\end{aligned}$$

Dividing by $L_{n,n+k}(1) = k(n+k)^{n-1}$ and setting $m = n+k$, we obtain the expected value.

**Proposition 7.3** *The expected value of $B(h)$ as $h$ varies over all hash functions from $\{1, 2, \ldots, n\}$ to $\{1, 2, \ldots, m\}$ $(n < m)$ is*

$$\frac{1}{2}\sum_{i=2}^{n}\binom{n}{i} i!\,m^{1-i} = \frac{1}{2}\left[\frac{n(n-1)}{m} + \frac{n(n-1)(n-2)}{m^2} + \cdots\right]. \blacksquare \tag{23}$$

To relate Proposition 7.3 to Knuth's results, we note that he considers the quantity $C'_{n-1}$ which is the expected number of probes to insert the $n$th object for a random hash function from $\{1, 2, \ldots n\}$ to $\{1, 2, \ldots m\}$. Since Knuth counts the probe of a vacant box, which we do not, (23) is equal to $\sum_{j=1}^{n}(C'_{j-1} - 1)$. Conveniently, he is also interested in the quantity $C_n = \frac{1}{n}\sum_{j=1}^{n} C'_{j-1}$. Since (23) is equal to $n(C_n - 1)$, it is easy to check that Knuth's formula (40) in [24, p. 530] agrees with Proposition 7.3.

# 8 Comments and open questions

Several areas related to what we have presented deserve further investigation.

(1) There are many other specializations of the Tutte polynomial that enumerate various classes of objects. See Brylawski's survey article [5] or his article with Oxley [6] for a list in the context of matroids. How many of these can be explained by either DFS?

(2) Stanley's interpretation of $t_G(2, 0)$ was actually part of a more general result [38]. He proved that if $G$ is a connected graph and $k$ is a positive integer, then

$$k\ t_G(1 + k, 0) \tag{24}$$

is the number of pairs $(\mathcal{O}, f)$ where

- $\mathcal{O}$ is an acyclic orientation of $G$, and
- $f : V \to \{1, 2, \ldots, k\}$ is a function such that $uv \in \mathcal{O}$. implies $f(u) \le f(v)$.

Also we know, from equation (8), that (24) counts pairs $(\mathcal{O}, g)$ where

- $\mathcal{O}$ is an acyclic orientation of $G$, and
- $g : V \to \{1, 2, \ldots, k\}$ is a function such that $u, v$ in the same initial component of $\mathcal{O}$ implies $g(u) = g(v)$.

Recently Serge Elnitsky [private communication] has found a direct bijection between such pairs.

(3) In Theorem 6.7, we proved an identity for $J(u) = \sum_{n\ge 0} t_{n+1}(y)u^n/n!$. This is a special case of the fact [16] that

$$J(u)J(yu)\cdots J(y^k u) = \sum_{n\ge 0}(1 + y + \cdots + y^k)^n y^{\binom{n}{2}} J(u)J(yu)\cdots J(y^{n-1}u)\frac{u^n}{n!}.$$

Unfortunately, we have not been able to find a combinatorial proof of this formula based on counting externally active edges.

(4) Parking functions have been receiving a lot of attention recently because of their connection with a problem in representation theory. The Rearrangement Lemma shows that there is an action of permutations $\pi$ in the symmetric group $S_n$ on parking functions $p : \{1,\ldots,n\} \to \{1,\ldots,n+1\}$ given by

$$\pi p(i) = p(\pi^{-1} i).$$

Thus the set of parking functions can be made into an $S_n$-module which we denote by $P_n$.

Now consider the polynomial ring $R_n = \mathbf{C}[x_1, \ldots x_n, y_1, \ldots, y_n]$ where $\mathbf{C}$ is the complex numbers. Let $\pi \in S_n$ act on $q \in R_n$ *diagonally*, i.e.,

$$\pi q(x_1, \ldots x_n, y_1, \ldots, y_n) = q(x_{\pi 1}, \ldots x_{\pi n}, y_{\pi 1}, \ldots, y_{\pi n}).$$

If $J \subseteq R_n$ is the ideal of nonconstant invariants of this action, then the quotient $R_n/J$ is another $S_n$-module. Mark Haiman conjectured that there is an isomorphism

$$P_n \cong Q \otimes (R_n/J) \tag{25}$$

where $Q$ is a module for the sign representation. Moreover, since the bidegree of a polynomial in the $x$'s and $y$'s is preserved under the action of $S_n$, $R_n/J$ is a bigraded $S_n$-module. If we ignore the $y$-grading then (25) seems to be an isomorphism of $x$-graded $S_n$-modules, where the degree of a parking function $p$ is $B(p)$ as defined in Section 7.

There is a sizable amount of numerical evidence for this conjecture. However, it is still mysterious that two such differently defined objects should turn out to be isomorphic. For more information about this question, see [22].

# 9 Tables

We will now give tables for various quantities that we have studied.

Tables 1a and 1b contain the Tutte polynomials of the complete graphs $t_n(x, y)$ for $n \leq 8$. For $n \leq 3$, the polynomials are written in the usual format. For $4 \leq n \leq 8$, we let

$$t_n(x, y) = \sum_{i,j} t_{n,i,j} x^i y^j$$

and then display the coefficients in a rectangular matrix with the entry in row $j$ and column $i$ of the $n$th array being being $t_{n,i,j}$.

Table 2 gives the specializations $t_n(y) = t_n(1, y)$ which are also inversion enumerators for trees.

$$t_1(x,y) = 1$$
$$t_2(x,y) = x$$
$$t_3(x,y) = x + x^2 + y$$

$t_4(x,y)$ :

| $j\backslash i$ | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 0 | 2 | 3 | 1 |
| 1 | 2 | 4 | 0 | 0 |
| 2 | 3 | 0 | 0 | 0 |
| 3 | 1 | 0 | 0 | 0 |

$t_5(x,y)$ :

| $j\backslash i$ | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | 0 | 6 | 11 | 6 | 1 |
| 1 | 6 | 20 | 10 | 0 | 0 |
| 2 | 15 | 15 | 0 | 0 | 0 |
| 3 | 15 | 5 | 0 | 0 | 0 |
| 4 | 10 | 0 | 0 | 0 | 0 |
| 5 | 4 | 0 | 0 | 0 | 0 |
| 6 | 1 | 0 | 0 | 0 | 0 |

$t_6(x,y)$ :

| $j\backslash i$ | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 0 | 0 | 24 | 50 | 35 | 10 | 1 |
| 1 | 24 | 106 | 90 | 20 | 0 | 0 |
| 2 | 80 | 145 | 45 | 0 | 0 | 0 |
| 3 | 120 | 105 | 15 | 0 | 0 | 0 |
| 4 | 120 | 60 | 0 | 0 | 0 | 0 |
| 5 | 96 | 24 | 0 | 0 | 0 | 0 |
| 6 | 64 | 6 | 0 | 0 | 0 | 0 |
| 7 | 35 | 0 | 0 | 0 | 0 | 0 |
| 8 | 15 | 0 | 0 | 0 | 0 | 0 |
| 9 | 5 | 0 | 0 | 0 | 0 | 0 |
| 10 | 1 | 0 | 0 | 0 | 0 | 0 |

Table 1a: Tutte polynomials of complete graphs for $n \leq 6$

$t_7(x,y)$ :

| $j\backslash i$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 120 | 274 | 225 | 85 | 15 | 1 |
| 1 | 120 | 644 | 721 | 280 | 35 | 0 | 0 |
| 2 | 490 | 1225 | 700 | 105 | 0 | 0 | 0 |
| 3 | 945 | 1330 | 420 | 35 | 0 | 0 | 0 |
| 4 | 1225 | 1085 | 210 | 0 | 0 | 0 | 0 |
| 5 | 1260 | 756 | 84 | 0 | 0 | 0 | 0 |
| 6 | 1120 | 469 | 21 | 0 | 0 | 0 | 0 |
| 7 | 895 | 245 | 0 | 0 | 0 | 0 | 0 |
| 8 | 645 | 105 | 0 | 0 | 0 | 0 | 0 |
| 9 | 420 | 35 | 0 | 0 | 0 | 0 | 0 |
| 10 | 245 | 7 | 0 | 0 | 0 | 0 | 0 |
| 11 | 126 | 0 | 0 | 0 | 0 | 0 | 0 |
| 12 | 56 | 0 | 0 | 0 | 0 | 0 | 0 |
| 13 | 21 | 0 | 0 | 0 | 0 | 0 | 0 |
| 14 | 6 | 0 | 0 | 0 | 0 | 0 | 0 |
| 15 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

$t_8(x,y)$ :

| $j\backslash i$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 720 | 1764 | 1624 | 735 | 175 | 21 | 1 |
| 1 | 720 | 4488 | 6020 | 3136 | 700 | 56 | 0 | 0 |
| 2 | 3444 | 10696 | 8470 | 2380 | 210 | 0 | 0 | 0 |
| 3 | 7980 | 15120 | 7490 | 1260 | 70 | 0 | 0 | 0 |
| 4 | 12495 | 15855 | 5320 | 560 | 0 | 0 | 0 | 0 |
| 5 | 15400 | 13986 | 3360 | 224 | 0 | 0 | 0 | 0 |
| 6 | 16261 | 11123 | 1960 | 56 | 0 | 0 | 0 | 0 |
| 7 | 15464 | 8196 | 980 | 0 | 0 | 0 | 0 | 0 |
| 8 | 13600 | 5580 | 420 | 0 | 0 | 0 | 0 | 0 |
| 9 | 11200 | 3500 | 140 | 0 | 0 | 0 | 0 | 0 |
| 10 | 8680 | 1988 | 28 | 0 | 0 | 0 | 0 | 0 |
| 11 | 6328 | 1008 | 0 | 0 | 0 | 0 | 0 | 0 |
| 12 | 4333 | 448 | 0 | 0 | 0 | 0 | 0 | 0 |
| 13 | 2779 | 168 | 0 | 0 | 0 | 0 | 0 | 0 |
| 14 | 1660 | 48 | 0 | 0 | 0 | 0 | 0 | 0 |
| 15 | 916 | 8 | 0 | 0 | 0 | 0 | 0 | 0 |
| 16 | 462 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 17 | 210 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 18 | 84 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 19 | 28 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 20 | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 21 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Table 1b: Tutte polynomials of complete graphs for $n = 7, 8$

$$t_1(y) = 1$$

$$t_2(y) = 1$$

$$t_3(y) = 2 + y$$

$$t_4(y) = 6 + 6\,y + 3\,y^2 + y^3$$

$$t_5(y) = 24 + 36\,y + 30\,y^2 + 20\,y^3 + 10\,y^4 + 4\,y^5 + y^6$$

$$\begin{aligned} t_6(y) = {} & 120 + 240\,y + 270\,y^2 + 240\,y^3 + 180\,y^4 + 120\,y^5 + 70\,y^6 + 35\,y^7 \\ & +15\,y^8 + 5\,y^9 + y^{10} \end{aligned}$$

$$\begin{aligned} t_7(y) = {} & 720 + 1800\,y + 2520\,y^2 + 2730\,y^3 + 2520\,y^4 + 2100\,y^5 + 1610\,y^6 + 1140\,y^7 \\ & +750\,y^8 + 455\,y^9 + 252\,y^{10} + 126\,y^{11} + 56\,y^{12} + 21\,y^{13} + 6\,y^{14} + y^{15} \end{aligned}$$

$$\begin{aligned} t_8(y) = {} & 5040 + 15120\,y + 25200\,y^2 + 31920\,y^3 + 34230\,y^4 + 32970\,y^5 + 29400\,y^6 \\ & +24640\,y^7 + 19600\,y^8 + 14840\,y^9 + 10696\,y^{10} + 7336\,y^{11} + 4781\,y^{12} \\ & +2947\,y^{13} + 1708\,y^{14} + 924\,y^{15} + 462\,y^{16} + 210\,y^{17} + 84\,y^{18} + 28\,y^{19} \\ & +7\,y^{20} + y^{21} \end{aligned}$$

Table 2: The inversion enumerator for trees

## References


[1] G. E. Andrews, Catalan numbers, $q$-Catalan numbers and hypergeometric series, *J. Combin. Theory Ser. A*, **44** (1987), 267–273.

[2] R. A. Bari, Chromatic polynomials and the internal and external activities of Tutte, in "Graph Theory and Related Topics (Proc. Int. Conf. Graph Theory Comb., University of Waterloo, 1977)," A. Bondy and U. S. R. Murty, eds., Academic Press, New York (1979), 41–52.

[3] J. S. Beissinger, On external activity and inversions in trees, *J. Combin. Theory Ser. B* **33** (1982), 87–92.

[4] A. Björner, The homology and shellability of matroids and geometric lattices, Chapter 7 in "Matroid Applications," N. White ed., Cambridge University Press, Cambridge, 1991, 226–283.

[5] T. Brylawski, The Tutte polynomial I: General theory, in "Matroid Theory and its Applications (Proc. C. I. M. E., Varenna, 1980)," Naples (1982), 125–276.

[6] T. Brylawski and J. Oxley, The Tutte polynomial and its applications, Chapter 6 in "Matroid Applications," N. White ed., Cambridge University Press, Cambridge, 1991, 123–225.

[7] W. F. Clocksin and C. S. Mellish, "Programming in Prolog," Springer-Verlag, Berlin, 1981.

[8] H. H. Crapo, The Tutte polynomial, *Aequationes Math.* **3** (1969), 211–229.

[9] J. E. Dawson, A construction for a family of sets and its application to matroids, in "Combinatorial Mathematics VIII," K. L. McAvaney ed., Lecture Notes in Math., Vol. 884, Springer-Verlag, New York, NY, 1981, 136–147.

[10] J. E. Dawson, A collection of sets related to the Tutte polynomial of a matroid, in "Graph Theory Singapore 1983," K. M. Koh and H. P. Yap eds., Lecture Notes in Math., Vol. 1073, Springer-Verlag, New York, NY, 1984, 193–204.

[11] P. Doubilet, G.-C. Rota and R. P. Stanley, On the foundations of combinatorial theory VI: The idea of generating function, in "Sixth Berkeley Symposium on Mathematical Statistics and Probability," (1972), 267–318.

[12] D. Foata, "La Série Génératrice Exponentielle dans les Problèmes d'Enumération," Séminaire de Mathématique Supérieurs, No. 54, Presses de l'Université de Montréal, Montréal, 1974

[13] D. Foata and J. Riordan, Mappings of acyclic and parking functions, *Aequationes Math.* **10** (1974), 10–22.

[14] J. Fürlinger and J. Hofbauer, $q$-Catalan numbers, *J. Combin. Theory Ser. A*, **40** (1985), 248–264.

[15] A. M. Garsia and D. Stanton, Group actions on Stanley-Reisner rings and invariants of permutation groups, *Adv. in Math.* **51** (1984), 107–201.

[16] I. M. Gessel, A noncommutative generalization and $q$-analog of the Lagrange inversion formula, *Trans. Amer. Math. Soc.*, **257** (1980), 455–482.

[17] I. M. Gessel, Enumerative applications of a decomposition for graphs and digraphs, *Discrete Math.*, to appear.

[18] I. M. Gessel and D.-L. Wang, Depth-first search as a combinatorial correspondence, *J. Combin. Theory Ser. A*, **26** (1979), 308–313.

[19] I. M. Gessel, B. E. Sagan and Y.-N. Yeh, Enumeration trees by inversions, *J. Graph Theory*, to be published.

[20] G. Gordon and L. Traldi, Generalized activities and the Tutte polynomial, *Discrete Math.* **85** (1990), 167–176.

[21] C. Greene and T. Zaslavsky, On the interpretation Whitney numbers through the arrangements of hyperplanes, zonotopes, non-Radon partitions, and orientations of graphs, *Trans. Amer. Math. Soc.*, **280** (1983), 97–126.

[22] M. Haiman, Conjectures on the quotient ring of diagonal invariants, *J. Algebraic Combin.*, **3** (1994), 17–76.

[23] D. J. Kleitman and K. J. Winston, Forests and score vectors, *Combinatorica* **1** (1981), 49–54.

[24] D. E. Knuth, "The Art of Computer Programming, Vol. 3: Sorting and Searching," 1973; Addison-Wesley, Reading, MA.

[25] D. E. Knuth, Computer science and its relation to mathematics, *Amer. Math. Monthly* **81** (1974), 323–342.

[26] A. G. Konheim and B. Weiss, An occupancy discipline and applications, *SIAM J. Appl. Math.* **14** (1966), 1266–1274.

[27] C. Krattenthaler, Counting lattice paths with linear boundary II, *Österreich. Akad. Wiss. Math.-Natur. Kl. Sitzungsber. II*, **198** (1989), 171–199.

[28] G. Kreweras, Une famille de polynômes ayant plusieurs propriétés énumeratives, *Period. Math. Hungar.*, **11** (1980), 309–320.

[29] V. A. Liskovets, On the number of maximal vertices of a random acyclic digraph, *Theory Probab. Appl.* **20** (1975) 401–409.

[30] C. L. Mallows and J. Riordan, The inversion enumerator for labeled trees, *Bull. Amer. Math. Soc.*, **74** (1968), 92–94.

[31] P. Moszkowski, Arbres et suites majeures, *Period. Math. Hungar.* **20 (2)** (1989), 147–154.

[32] A. Renyi, On connected graphs I, *Publ. Math. Inst. Hungar. Acad. Sci.*, **4** (1959), 385–388.

[33] J. Riordan, "Combinatorial Identities," Wiley, New York, NY, 1968.

[34] J. Riordan, Ballots and trees, *J. Combin. Theory* **6** (1969), 408–411.

[35] R. W. Robinson, Counting labeled acyclic digraphs, in "New Directions in the Theory of Graphs," F. Harary, ed., Academic Press, New York, 1973.

[36] V. I. Rodionov, On the number of labeled acyclic graphs, *Discrete Math.* **105** (1992), 319–321.

[37] M. P. Schützenberger, On an enumeration problem, *J. Combin. Theory* **4** (1968), 219–221.

[38] R. P. Stanley, Acyclic orientations of graphs, *Discrete Math.*, **5** (1973), 171–178.

[39] R. P. Stanley, Exponential structures, *Studies Appl. Math.*, **59** (1978), 73–82.

[40] W. T. Tutte, A contribution to the theory chromatic polynomials, *Canad. J. Math.* **6** (1953), 80–91.

[41] W. T. Tutte, On dichromatic polynomials, *J. Combin. Theory* **2** (1967), 301–320.

[42] H. S. Wilf, "Generatingfunctionology," Academic Press, Boston, MA, 1990.

[43] E. M. Wright, The number of connected sparsely edged graphs, *J. Graph Theory*, **1** (1977), 317–330.