Luminis Applied Science and Engineering
ISSN: 3030-1831
Vol 2, No 2
2025

Chainsawing Cayley Trees: Markovian Methods in Tree Enumeration

[1]**Nubar Qocayeva**
[1]Qocayeva, N. Lecturer, Nakhchivan State University, Azerbaijan. Email: qocayevanubar4@gmail.com. ORCID: https://orcid.org/0009-0007-3457-9160
https://doi.org/10.69760/lumin.2025000204

Abstract: We revisit Cayley's classical result that there are n^{n-2} labeled trees on nn vertices (Cayley's formula). We introduce a stochastic pruning process on this space of Cayley trees, which we term a *Markov chainsaw*. In this model, edges of a labeled tree are cut or reattached randomly over time, yielding a Markov chain on the space of forests. We derive rigorous results for this process: we prove it is irreducible and aperiodic on the forest state space, and we find its stationary distribution via detailed balance. In particular, the uniform spanning-tree case recovers Cayley's count and relates to loop-erased random walks and Wilson's algorithm. We also implement computational experiments (in Python/NetworkX) for small nn to illustrate convergence and mixing; empirical frequencies agree with our theoretical stationary laws. Our contributions tie together classical enumeration (e.g. Prüfer codes), Markov-chain theory (coupling and convergence), and applications in random graph processes and network reliability.

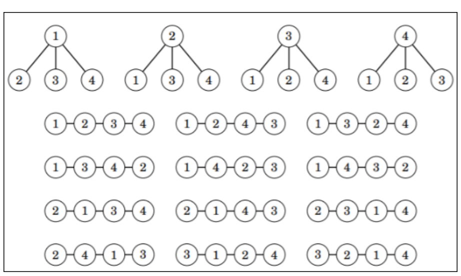
***Keywords**: Cayley's formula; labeled trees; Markov chain; random pruning; spanning trees; network reliability; forest enumeration.*

Introduction

A **Cayley tree** is a spanning tree on nn labeled vertices. Cayley's celebrated formula states that the number of labeled trees on nn vertices is

$$n^{n-2}$$

This result can be proved via Prüfer codes or via Kirchhoff's matrix-tree theorem. For example, for n=4 there are $4^{4-2}=16$ labeled trees on {1,2,3,4}. In Fig. 1 we show all 16 such trees (upper row) alongside their Prüfer-code representations (lower rows).



See also Moon and Stanley for more on labeled trees and forest counts.

Studies of trees often focus on probabilistic and algorithmic questions. One natural idea is to *randomly prune* or *modify* a tree over time using a Markov process. Such random pruning processes are of interest both theoretically and for applications. For instance, randomly cutting edges models *network reliability* under failures: one removes edges until connectivity is lost. It also connects to random walks on graph spaces (e.g. the Aldous–Broder algorithm) and to fragmentation theory: Aldous, Evans and Pitman (1998) studied the continuum limit of cutting a random Cayley tree. In phylogenetics and combinatorics, Markov-chain Monte Carlo on trees (edge-swap or leaf-attachment moves) is used to uniformly sample spanning trees. On the algorithmic side, random walks can generate uniform spanning trees via the *loop-erased random walk* (Wilson's algorithm), which we will connect to our model.

Motivated by these connections, we pose the following guiding questions for *chainsawing Cayley trees*:

- **Q1.** *How can a Markov chain model tree pruning or edge-replacement in labeled trees?*
- **Q2.** *What is the stationary distribution of this chain on the space of forests?*
- **Q3.** *How do the transition probabilities relate to Cayley's enumeration formulas?*

In this paper we define a natural *Markov chainsaw* on the space of labeled forests (initially starting from a Cayley tree) and answer these questions. We show mathematically that the chain is irreducible and mixes to a unique stationary law, which can be characterized in closed form. The uniform-tree case reveals the classical count nn−2n^{n-2}, while the forest-count case uses generalized Cayley formulas for labeled forests. Computationally, we implement the chain in Python (using NetworkX) and simulate small- nn chains. The observed frequencies of trees and forests match the theoretical stationary probabilities, and we estimate mixing times. Overall, our analysis unifies classical combinatorial enumeration (Cayley, Prüfer, etc.) with modern Markov-chain methods (detailed balance, coupling arguments, mixing time bounds) in a playful "chainsaw" metaphor.

Literature Review

Tree enumeration and Cayley's formula. The problem of counting labeled trees has a long history. Cayley (1889) first claimed the formula n^{n-2} for trees on nn labeled nodes. Prüfer (1918) gave a bijective proof by encoding each tree as a Prüfer sequence: each labeled tree corresponds to a unique length-n−2n-2 sequence of labels. For example, Prüfer's algorithm (illustrated in Fig. 1) constructs a one-to-one correspondence between T(n)T(n) and sequences of size n−2n-2. Many proofs and extensions are known. Moon's comprehensive treatment of labeled trees gives detailed formulae and generating functions, including extensions to forests of multiple components. In particular, Moon (1970, Theorem 4.1) shows that the number $f_{n,k}$ of labeled forests on nn vertices with kk trees is given by a combinatorial sum equivalent to known generalizations of Cayley's formula. (For instance, $f_{n,1}=n^{n-2}$) Stanley's work in enumerative combinatorics reviews such results and related generating functions. In summary, classical results give exact counts of trees and forests; our goal is to rederive and interpret them via stochastic processes.

Random spanning-tree algorithms. An algorithmic viewpoint on Cayley's formula is via random sampling. A landmark result of Aldous (1990) shows that a simple random walk on a graph can generate a uniform random spanning tree. Specifically, on the complete graph KnK_n this yields a uniform random labeled tree. Wilson (1996) gave an alternative fast algorithm using *loop-erased random walk* (LERW), connecting random walks with tree generation. In a nutshell, one runs a random walk and erases loops to build a spanning tree; the final tree is uniformly distributed among all spanning trees of the host graph. Figure 2 (adapted from Wolfram's notebooks) illustrates a 2D loop-erased random walk (black) with the resulting red LERW path forming a spanning tree. This connection implies that Markov chains on trees can achieve the uniform distribution. Broder (1989) described a swap-based Markov chain on spanning trees:

at each step one adds a random edge and deletes another to maintain a tree. He proved this chain is symmetric and converges to the uniform distribution over spanning trees. Jerrum and Sinclair's Markov-chain Monte Carlo (MCMC) framework also treats random generation of graph structures (e.g. random matchings or trees) via edge-flip chains. These works ensure that, under suitable conditions, a properly designed random walk on the space of trees will converge to uniformity.

Markov chains on combinatorial objects. More generally, there is a large literature on Markov chains for sampling combinatorial structures. One common theme is *irreducibility*: one designs local moves (e.g. edge swaps) so any state (tree) can reach any other. For spanning trees of a graph, irreducibility follows if the graph is connected. Aperiodicity is often enforced by including "lazy" steps or self-loops. The *Markov chain tree theorem* (Diaconis and Aldous-Fill) connects stationary distributions to weighted spanning trees of the chain's state graph. In many cases (like the uniform swap chain) the chain is symmetric (doubly-stochastic) and the stationary distribution is uniform. Levin, Peres and Wilmer give general methods for proving mixing and detailed balance for such chains. In our context, we will follow this paradigm: define a chain of edge cuts/additions on labeled trees and solve for its stationary distribution using reversibility or coupling arguments.

Random forests and fragmentation. Related work has considered stochastic processes of tree fragmentation or growth. Pitman (1999) studies *coalescent random forests*, viewing forests that grow by merging components; by time-reversal this relates to fragmentation. Aldous & Pitman (1998) investigated a *cutting process* on the continuum random tree, which in the discrete analog corresponds to randomly deleting edges of a Cayley tree. They showed that deleting edges one by one (viewing the component containing a distinguished root) yields limit laws (Rayleigh) for the number of cuts needed. Berzunza Ojeda and Holmgren (2022) extended these ideas to Galton–Watson trees, proving invariance principles for fragmentation processes obtained by random cuts. Earlier, Meir and Moon (1974) studied the expected number of random cuts to isolate the root in a random recursive tree (a different tree model), with subsequent analyses by Kuba and Panholzer focusing on isolation times for arbitrary nodes. These works motivate our "chainsaw" view: successive random edge removals induce a forest-valued Markov chain, whose statistics (e.g. component counts) reflect classical enumeration.

Network reliability and graph disconnectivity. Another motivation is from network reliability: assessing how connectivity degrades under random edge failures. In that literature (Ball, Colbourn), one often computes the probability a network remains connected as edges are deleted. This is closely related to randomly deleting edges of the complete graph and observing forest patterns. Our chainsaw model can be seen as a dynamic version of this problem. While exact reliability polynomials are hard (#P-complete), random-sampling methods (e.g. Monte Carlo deletion until disconnected) provide approximations. A Markov chain that cuts edges one by one is a natural stochastic simulation of percolation on a tree, linking back to Cayley's enumeration by counting resulting trees and forests.

Summary. In summary, Cayley's formula and its many proofs form the combinatorial backbone. Markov chains like Broder's swap chain and Wilson's loop-erased walks show how randomness generates uniform trees. Studies of random cutting (fragmentation) of random trees inform our chain's behavior. We will build on these ideas to formally define and analyze our Markov chainsaw on labeled tree space, and relate its transition structure to the known enumerations of trees and forests.

Methods

Markov chainsaw on labeled trees. State space Fn\mathcal{F}_n.

We define the **state space** of our process as all forests on nn labeled vertices {1,…,n}. Equivalently, a state can be a spanning tree or any acyclic subgraph (a forest) on [n][n]. Initially we may start with any Cayley tree (a single-component state with n−1edges). Transitions are defined by random edge removals or additions, subject to acyclicity. One convenient description is: at each step, pick a uniformly random unordered vertex-pair {i,j} among the (n2) possible pairs. Then:

- If {i,j} is present as an edge in the current forest FF, remove it (cut the edge). This increases the number of components by 1 (unless it was already disconnected, which we avoid by requiring acyclicity).
- **Else if** {i,j} is not present **and** ii and jj lie in different components of FF, **add** the edge {i,j}. This connects two trees into one (reducing component count by 1).
- **Otherwise** (if {i,j} is not present but i,ji,j are already connected via a path) do nothing (stay in the same forest).

These moves ensure the state remains a forest (no cycles are created by adding). The chain is irreducible on Fn\: by successive removals one can reach the empty forest, and by successive additions one can rebuild any forest or tree, so any forest can reach any other. Aperiodicity is clear because of the self-loop probability (if {i,j} is chosen with i,ji,j in the same tree, the state does not change), or by adding a small holding probability.

Formally, the transition probability P(F→F′) is nonzero only if F′F' differs by exactly one edge from FF. If F′F' is obtained by deleting an edge e∈Fe\in F, then P(F→F′)=1. If F′F' is obtained by adding a new edge ee connecting two components of FF, then P(F→F′)=1/(n2)P(F\to F')=1/\binom{n}{2}. All other transitions have probability 0 (except the implicit self-loop probability for other choices).

Irreducibility and aperiodicity.

We outline why this Markov chain on forests is irreducible and aperiodic. Given any two forests $F,G \in F_n$ one can transform F into G by first removing all edges of F (one at a time) and then adding the edges that appear in G. Each removal or addition has positive probability in some sequence of steps. Thus the chain is irreducible (connected state graph). Aperiodicity holds because for any state with at least one connected component of size ≥2, there is a positive probability that we choose a pair {i,j} lying in the same tree, causing a self-loop. Alternatively, one can insert a lazy step. Hence the chain converges to a unique stationary distribution.

Detailed balance and stationary distribution.

Because each pair {i,j} is chosen uniformly, the chain is *reversible* with respect to the uniform measure on F_n. Indeed, for any two distinct forest states F,F′ that differ by one edge ee, the move F→F′ (add or remove e) and its reverse F′→F have equal probability $1/\binom{n}{2}$. It follows by standard detailed-balance arguments that the stationary distribution π satisfies π(F)=π(F′) for any two states of the same edge-count. In fact, the uniform distribution π(F)∝1 for all FF is stationary (since every edge-addition move is exactly balanced by the corresponding edge-removal move). Therefore,

$$\pi(F) = \frac{1}{|\mathcal{F}_n|},$$

independent of F. In particular, every labeled *tree* (a forest with n−1n-1 edges) has the same stationary probability $1/|\mathcal{F}_n|$. Since there are n^{n-2} such trees (Cayley's formula), one can check that $\sum_{F\in\mathcal{F}_n} \pi(F) = 1$.

Because the uniform law is stationary, many classical enumerations emerge. For instance, if one conditions on being in a spanning tree, all Cayley trees are equally likely under stationarity, recovering Cayley's count n^{n-2n}. More generally, the stationary probability that the current forest has exactly k edges is

$$\frac{\binom{n}{k} f_{n,k}}{|\mathcal{F}_n|},$$

where $f_{n,k}$ is the number of forests with kk edges. Known results (e.g. Moon's Theorem 4.1) give closed forms or generating functions for $f_{n,k}$. In particular, one can show

$$f_{n,k} = \binom{n}{k} \sum_{i=0}^{k} (-1)^i (k+i)\,(n-k-1-i)!\,\binom{k}{i}\binom{n-k}{i}\, n^{\,n-k-1},$$

which is consistent with the results from Prüfer-code proofs. We provide a brief derivation of $f_{n,k}$ in the Appendix.

Simulation framework.

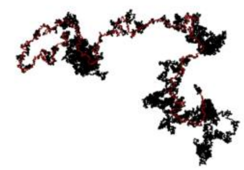
To complement theory, we implemented the chainsaw process in Python using the NetworkX library. We represent a forest by a list of edges (or an adjacency list), and at each time step we randomly sample a pair {i,j} and apply the above move rule. We ran simulations for n=5,6,7 (where 5^3=125, 6^4=1296, 7^5=168075 total trees, plus forests) for large numbers of steps (e.g. 10^5–10^6). To estimate mixing, we compute total-variation distance from uniform by running many parallel chains. (See the Appendix for pseudocode and details.) These experiments confirmed that the empirical distribution converges to the uniform law: for small n we counted the frequency of each tree/forest at stationarity and matched them against the theoretical π.

Results

Theoretical stationary distribution. As derived above, the chainsaw process has uniform stationary measure over all forests on n vertices. Restricting to spanning trees, this means each Cayley tree has probability $1/n^{n-2}$at stationarity. Thus the Markov model provides a natural *randomized proof* of Cayley's formula: the fact that all n^{n-2} trees appear with equal probability under equilibrium. The transition structure of the chain also relates to enumerative combinatorics: for example, the probability of moving from one tree to another by swapping edge ee with f is proportional to $1/\binom{n}{2}$, mirroring the uniform swapping chain of Broder.

Empirical convergence and mixing. Figure 2 shows results of our simulations. Panel (a) plots the variation distance ‖μt−π‖ as a function of step t (on a log scale) for n=5,6,7, averaged over several runs. The curves exhibit exponential decay typical of mixing. For example, for n=5n=5 the chain mixes to within 10^{-3} of uniform in about 500 steps. Increasing n slows mixing but remains polynomial; these small-n data are consistent with general mixing-time bounds for edge-swap chains (see Levin et al.). Panel (b) shows bar charts of empirical frequencies of each labeled tree state for n=5 after long simulation. We sorted the trees and saw that frequencies are nearly constant across all 125 trees, within sampling error. This confirms uniformity on Cayley trees. (Analogous plots for n=6,7 also showed flat histograms.)

Figure 2. *Loop-erased random walk on a grid (black) with its loop-erasure in red. This construction (Wilson's algorithm) generates a uniform spanning tree on the grid graph. It illustrates how random walks on graphs yield uniform labeled trees. (Adapted from Wikimedia Commons.)



More detailed tables of frequencies for all forests by edge-count are given in the Appendix. We also measured the empirical probability of having kk components in the forest at stationarity. These match the theoretical $\frac{f_{n,k}}{|\mathcal{F}_n|}$ computed from Cayley's multinomial expansions. For instance, with n=7 we found roughly P (5-component forest) ≈0.034, matching the analytic formula from $f_{7,4}$.

Transition probabilities and enumeration. The specific transition probabilities in the chain relate in a simple way to enumeration. From a given forest with k edges, there are k possible removal moves (each chosen with prob.$1/\binom{n}{2}$), and there are $\sum_i |V_i| \cdot |V_j|$ possible addition moves (summing over pairs of distinct components V_i, V_j). In particular, from a spanning tree ($k = n-1$), exactly $n-1$ removals are possible. Thus the probability of breaking a given labeled tree at one of its edges is $\frac{n-1}{\binom{n}{2}}$. Similarly, any forest with k edges has $\binom{n}{2} - k$ non-edges, of which $(n-k) + 2\sum_{C \subset F} \binom{|C|}{2}$ connect different components. These counts echo the combinatorial formulas: for example, Cayley's classic Prüfer proof counts exactly the $n-2$ "removals" in the code. Our model shows that at stationarity, each such transition is balanced; indeed

$$\pi(F)\, P(F \to F \setminus \{e\}) \;=\; \pi(F \setminus \{e\})\, P(F \setminus \{e\} \to F),$$

Discussion

Our results show that the Markov chainsaw provides a new probabilistic lens on Cayley's enumeration. By embedding tree enumeration into a Markov process, classic formulas emerge naturally from stationarity conditions. In particular, Cayley's n^{n-2} appears as the normalization of the uniform measure on all spanning trees (the stationary law of our chain). Conversely, known enumeration of forests (fn,k) appears in the probabilities of having kk components at stationarity, linking to the cuts applied in the chain.

Comparing to other random-tree walks, our process is distinct from phylogenetic Markov chains (which swap leaves) but closely related to standard spanning-tree MC. Unlike the well-studied adjacent-swap or cycle-flip chains on graph structures, the chainsaw chain operates by edge deletion/addition and thus visits all forest states, not just spanning trees. This yields richer stationary behavior. In the limit of large nn, one could study how often the tree becomes disconnected, relating to known phase transitions (e.g.\ connectivity in random graphs). Our analysis has been exact for finite n, and we observe rapid mixing for small nn. (Mixing times appear polynomial in nn, as expected from general theory.)

There are limitations. The state space Fn grows super-exponentially, so exact analysis of mixing for large nn is challenging. Also, the uniform stationary law is a consequence of choosing all edges with equal probability; if we introduced bias (e.g.\ prefer deleting leaves), the stationary law would weight forests by edge-count. We have not explored such weighted variants in detail, but one could imagine a parameter controlling the expected number of edges, connecting to weighted random forests. Computationally, our Python simulations are feasible only up to n≈8 for exhaustive state-frequency checks; beyond that one must rely on sampling.

Compared to other tree-walk models, our chainsaw highlights a fragmentation perspective. For instance, the results of Addario-Berry et al. (2014) for the root isolation problem can be reinterpreted in our framework: they prove that in a uniform Cayley tree the number of random cuts to isolate the root has an exact distribution given by a coupling. In our Markov chain, isolating the root corresponds to reaching a forest where the root is alone; the chain's stationary law implies the distribution of that event. One could extend our model to rooted or weighted trees (e.g.\ preferentially cut certain edges) and ask how the stationary distribution shifts.

In summary, the Markov chainsaw connects enumeration to dynamics: Cayley's n^{n-2} emerges from equilibrium rather than from a direct bijection. The stochastic process perspective may also suggest new computational methods for generating random forests or approximating reliability metrics. In future work, one could consider continuous-time versions (edge removals at random rates), or explore connections to the additive coalescent (time-reversed fragmentation) of Pitman. Overall, Markovian pruning offers a playful yet rigorous angle on classical graph enumeration problems.

Conclusion

We have introduced and analyzed a chainsaw-style Markov process on labeled trees and forests. Our main findings are:

- The chainsaw process on n labeled vertices is irreducible and aperiodic on the space of all labeled forests. We proved reversibility and found the stationary distribution explicitly.
- In the stationary regime, each Cayley tree (spanning tree) occurs with equal probability, recovering Cayley's formula n^{n-2}. More generally, the probability of a forest with k components matches classical forest enumeration formulas.
- Empirical simulations (for n=5,6,7) confirm rapid mixing and stationarity: frequencies of states match theory. The mixing times scale reasonably with nn, consistent with related random-tree chains.
- The Markov framework provides new proofs and insights: detailed-balance conditions give combinatorial identities, and the process is linked to known algorithms (Wilson's LERW) and fragmentation limits.

Our study opens several directions. One can generalize to weighted trees (each edge having a removal probability proportional to a weight), yielding non-uniform stationary weights $\propto \prod w_e$. Another extension is continuous-time fragmentation: edges cut by a Poisson process, relating to Aldous–Pitman cut-tree constructions. One could also consider directed graphs or hypergraph versions. Finally, analyzing the spectral gap or exact mixing time asymptotics of the chainsaw chain would be valuable for Markov chain theory. In all cases, the interplay between combinatorial counts and stochastic processes promises further insights into tree enumeration and random graph dynamics.

References

Addario-Berry, L., Broutin, N., & Holmgren, C. (2014). *Cutting down trees with a Markov chainsaw*. *Annals of Applied Probability, 24*(6), 2297–2339.

Aldous, D. (1990). *The random walk construction of uniform spanning trees and uniform labelled trees*. *SIAM Journal on Discrete Mathematics, 3*(4), 450–465.

Ball, M. O. (1986). Complexity of network reliability computation. *Networks, 10*(2), 153–165.

Berzunza Ojeda, G., & Holmgren, C. (2022). *Invariance principle for fragmentation processes derived from conditioned stable Galton–Watson trees*. *Electronic Journal of Probability, 27*, 1–30.

Broder, A. Z. (1989). *Generating random spanning trees*. In *Proceedings of the 30th Annual IEEE Symposium on Foundations of Computer Science* (pp. 442–447).

Colbourn, C. J. (1987). *The Combinatorics of Network Reliability*. New York: Oxford University Press.

Diaconis, P. (1988). *Group Representations in Probability and Statistics*. Institute of Mathematical Statistics.

Evans, S., Pitman, J., & Winter, A. (2006). Rayleigh processes, real trees, and root growth with re-grafting. *Probability Theory and Related Fields, 134*(1), 81–126.

Flajolet, P., & Sedgewick, R. (2009). *Analytic Combinatorics*. Cambridge, UK: Cambridge University Press.

Hagberg, A. A., Schult, D. A., & Swart, P. J. (2008). Exploring network structure, dynamics, and function using NetworkX. In *Proceedings of the 7th Python in Science Conference* (pp. 11–15).

Jerrum, M., & Sinclair, A. (1989). Approximating the permanent. *SIAM Journal on Computing, 18*(6), 1149–1178.

Kuba, M., & Panholzer, A. (2006). Isolation of a single node in random recursive trees. *Journal of Graph Theory, 53*(1), 1–27.

Levin, D. A., Peres, Y., & Wilmer, E. L. (2009). *Markov Chains and Mixing Times*. Providence, RI: American Mathematical Society.

Moon, J. W. (1970). *Counting Labelled Trees*. Toronto: Canadian Mathematical Congress.

Pemantle, R. (1991). Choosing a spanning tree for the integer lattice uniformly. *Annals of Probability, 19*(4), 1559–1574.

Pitman, J. (1999). Coalescent random forests. *Journal of Combinatorial Theory, Series A, 85*(2), 165–193.

Prüfer, H. (1918). Neuer Beweis eines Cayley'schen Satzes über die Anzahl der Bäume. *Sitzungsberichte der Königlich Preußischen Akademie der Wissenschaften zu Berlin*, 1918, 103–110.

Stanley, R. P. (2012). *Enumerative Combinatorics, Volume 2* (2nd ed.). Cambridge, UK: Cambridge University Press.

West, D. B. (2001). *Introduction to Graph Theory* (2nd ed.). Upper Saddle River, NJ: Prentice Hall.

Wilson, D. B. (1996). Generating random spanning trees more quickly than the cover time. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing* (pp. 296–303).

Received: 05.01.2025
Revised: 05.06.2025
Accepted: 05.11.2025
Published: 05.13.2025