An Intimate Analysis of Cuckoo Hashing with a Stash

Daniel Noble

April 6, 2021

Abstract

Cuckoo Hashing is a dictionary data structure in which a data item is stored in a small constant number of possible locations. It has the appealing property that the data structure size is a small constant times larger than the combined size of all inserted data elements. However, many applications, especially cryptographic applications and Oblivious RAM, require insertions, builds and accesses to have a negligible failure probability, which standard Cuckoo Hashing cannot simultaneously achieve. An alternative proposal introduced by Kirsch et al. is to store elements which cannot be placed in the main table in a "stash", reducing the failure probability to $\mathcal{O}(n^{-s})$ where n is the table size and s any constant stash size. This failure probability is still not negligible. Goodrich and Mitzenmacher showed that the failure probability can be made negligible in some parameter N when $n = \Omega(log^7(N))$ and $s = \Theta(logN)$. In this paper, I will explore these analyses, as well as the insightful alternative analysis of Aumüller et al. Following this, I present a tighter analysis which shows failure probability negligible in N for all $n = \omega(\log(N))$ (which is asymptotically optimal) and I present explicit constants for the failure probability upper bound.

1 Introduction

Cuckoo hashing is a hash table implementation that improves performance by allowing objects to be stored in a number of locations [PR01]. Pagh and Rodler discovered that this small modification greatly reduced the probability of a build failure. Specifically, a Cuckoo Hash table can store n elements of size $\log(N)$ in $\mathcal{O}(n\log(N))$ space, with failure probability $\Theta(\frac{1}{n})$ (see [DK12] for the explicit constant).

For many applications this failure probability was sufficient. In the case that a failure did occur, the hash table could simply be rebuilt with new hash functions. While a hash table rebuild requires $\Theta(n)$ computation, the probability of this occurrance is $\Theta(\frac{1}{n})$, so the *amortized* computation cost per access is still constant.

However, over the subsequent years, a number of applications arose that caused this failure probability to be insufficient. Specifically:

- For certain applications, the indexes were sensitive data, and as such a rebuild now constituted a security failure.

- Moreover, for some such applications the probability of a security failure needed to be *negligible* in some parameter 2^λ.

- Moreover, for some of those applications, n, the number of data items might be significantly smaller than (*e.g.,* polylogarithmic in) 2^λ.

A sequence of works arose to address these problems.

Firstly Kirsch, Mitzenmacher and Wieder explored the modification that any items which could not be stored in the Cuckoo Hash table would be stored in a"stash" of constant size s [KMW09]. They showed that the probability of a build failure was then reduced to $\mathcal{O}(n^{-s})$.

While this analyses allowed the failure probability to be reduced significantly, it only applied to constant s, so did not allow the failure probability to be negligible in n.

While Cuckoo Hashing was initially designed to allow for a constant number of accesses, in certain situations it was acceptable to have a super-constant number of stash accesses if this could provide a negligible build failure probability. For instance, since the stash is small, the stash may be stored in a lower memory level, so accesses to the stash may be significantly cheaper than accesses to the hash table. Another important use-case is Oblivious RAM. Oblivious RAM is a cryptographic primitive in which a trusted client can store data on an untrusted RAM. While encryption allows the client to hide the contents of the data, ORAM ensures that the client can also hide its access patterns from the untrusted RAM. Many Oblivious RAM designs use the Hierarchical approach, where data is stored in various hash tables of exponentially increasing sizes. The client may have a small amount of memory available itself, which may be enough for instance to store the stash. Therefore it is reasonable in this model for the cost of stash accesses to be counted separately from those of accesses to the Cuckoo Hash table(s). [1]

Goodrich and Mitzenmacher developed such an Oblivious RAM protocol [GM11]. While the analysis of Kirsch et al. did not provide negligible failure probability, Goodrich and Mitzenmacher showed how to extended this analysis to achieve negligigble failure probability in certain cases. They proved that, provided $n = \Omega(\log^7(N))$ a stash of size $s = \Theta(\log(N))$ would result in a failure probability negligible in N. (For tables of size $o(\log^7(N))$ another type of oblivious hashing data structure was needed.)

Aumüller, Dietzfelbinger and Woelfel then presented an elegant alternative analysis of Cuckoo Hashing with a Stash based on graph counting [ADW14].

[1] Other Hierachical ORAMs allowed the client to only have constant memory usage, and solved the problem of super-constant stashes by reinserting stash elements into another level, or having a single shared stash. This meant that only non-stash elements would be accessed in any given Hash Table, allowing for a constant number of accesses in these Hash Tables.

They showed firstly, that for constant s the probability of a build failure was further upper-bounded by $\mathcal{O}(n^{-(s+1)})$. They then showed that for sufficiently large n, the failure probability is $\mathcal{O}(n^{-\frac{s}{2}})$ when $s \leq n^{\frac{1}{3r}}$, for a suitable constant r.

These analyses leave open the question of how small n can be in terms of N and still have a Cuckoo Hash table with a stash with negligible probability of build failure. This survey reviews the analyses above in detail. It then extends the analysis to present a bound that is asymptotically tight. The bound on the failure probability also contains explicit constants. This is an important towards constructing concrete hierarchical ORAM implementations, which require concrete bounds for build failures of small cuckoo hash tables.

2 Notation

We will use the following notation and variables throughout the table:

$[x]$: The set of integers $1, \ldots, x$ for some integer $x > 0$.
n: The number of items to be stored in the table.
N: A parameter $N > n$ such that failure should be negligible in N.
m: The size of the regular hash tables. (We will examine the two-table case, so the total Cuckoo hash table will have size $2m$.)
$Po\,(\mu)$: The Poisson distribution, with parameter μ.
$Bin\,(M, p)$: The Binomial distribution, with M trials, each with probability of success p.

3 Cuckoo Hashing

Cuckoo Hashing in its simplest form involves 2 hash functions, h_1 and h_2, and 2 hash tables, T_1 and T_2, each with $m = \epsilon n$ locations of capacity 1. Each hash table has a unique hash function, and the hash functions are assumed to produce outputs uniformly at random in $[m]$. The tables consist of pairs (x, y) where x is the dictionary key and y is the dictionary value. An item (x, y) is stored in the table by being inserted into $T_1[h_1(x)]$. If another item (x', y') was stored in that location, it is removed from its original location (like a baby bird being displaced from its nest by a Cuckoo chick) and is placed in $T_2[h_2(x')]$. This may replace another item, which the algorithm likewise attempts to insert. This process continues either until every item has found a location in which to be inserted, or some threshold on the recursion depth is reached.[2] In the latter case the insertion has "failed". This triggers a "table rebuild" in which new tables are created with new hash functions and the algorithm attempts to insert every element into the new hash table.

[2] Many works (e.g. [KMW09]) set this recursion depth to $\alpha \log(N)$ for a sufficiently large constant α. This makes the probability that an item that can be inserted is not inserted small, but does not make this probabality negligible. Therefore we instead in our analysis assume that the maximum recursion depth is $2n$, which ensures an optimal allocation.

Cuckoo Hash tables can be generalized to have a larger number of hash functions. They can also be generalized to use a single table.

4 A Lower Bound

We begin by showing the following lower bound on the number of elements n in terms of the security parameter N, such that cuckoo hashing with a stash can fail with negligible probability in N. For consistency with other parts of the paper, we use the 2-table construction but this can easily be adapted to other constructions.

Theorem 1. *If $n = \mathcal{O}(\log(N))$ and $n - s = \Omega(n)$ then it is impossible for a 2-table Cuckoo Hash table to have a negligible build failure probability in N.*

Proof. Since $n - s = \Omega(n)$, it follows that $n - s \geq c_0 n$ for sufficiently large n where the constant c_0 satisfies $0 < c_0 \leq 1$. Therefore:

$$\begin{aligned}
\frac{n-s}{n} &\geq c_0 \\
\frac{n-s-2}{n} &\geq c_0 - \frac{2}{n} \\
\frac{n-s-2}{n} &\geq \frac{c_0}{2} \text{ when } n \geq \frac{4}{c_0} \\
\frac{n-s-2}{n} &\geq c_1 \text{ for constant } c_1 \text{ satisfying } 0 < c_1 \leq \frac{1}{2}
\end{aligned}$$

Since $n = \mathcal{O}(\log(N))$, there is some constant c_2 such that $n \leq c_2 \log(N)$ (for sufficiently large n).

Let $m = \epsilon n$ be the size of each table.

If all n items are hashed to the first $\lceil \frac{n-s-2}{2} \rceil$ locations in both tables, then $2\lceil \frac{n-s-2}{2} \rceil \leq n-s-1$ items can be stored in the table, and s items can be stored in the stash, but 1 item will not be able to be stored at all, so the build fails.

The probability that all n items are stored in the first $\lceil \frac{n-s-2}{2} \rceil$ locations in both tables is at least:

$$\begin{aligned}
\left(\frac{n-s-2}{2\epsilon n}\right)^{2n} &\geq \left(\frac{c_1}{2\epsilon}\right)^{2c_2\log(N)} \\
&\geq N^{2c_2\log\left(\frac{c_1}{2\epsilon}\right)}
\end{aligned}$$

This is non-negligible in N. Therefore the probability of a build failure is non-negligible.

□

This immediately implies the contrapositive:

Corollary 1. *Cuckoo Hashing with a stash requires $n - s = o(n)$ or $n = \omega(\log(N))$ in order to succeed with failure negligible in N.*

The case that $n - s = o(n)$ is very unnatural–it implies that a sub-constant number of elements are stored in the table, at which point the Cuckoo table is not providing much use. Thus, in any realistic setting where Cuckoo tables are used, it is necessary that $n = \omega(\log(N))$. This provides the lower bound for n in terms of N such that Cuckoo Hashing with a stash has a negligible probability of failure. We will later provide analysis that shows that this is also the *upper bound* for Cuckoo Hashing to succeed, so this bound is tight.

5 Graph Representation

Analyses of Cuckoo Hash table failure often represent the problem as a graph problem as follows. For each location in the Cuckoo hash table, create a vertex. Since the Cuckoo hash table has two tables each with capacity m, there will be $2m$ vertices. For each element stored in the Cuckoo hash table, draw an edge between the two locations in which it may be stored. Let G be the resulting graph. Since there will be one location from each table, G will be bipartite, with m vertices in each part. There may also be multiple edges between a pair of vertices, so G is a multigraph. Observe also that the graph is not connected: since $n < m$ some nodes will not be connected to any edges and there may also be multiple connected components that contain edges.

Let $\gamma(G)$ denote the cyclotomic number of G, that is the minimum number of edges that must be removed in order for G to be acyclic. Let $\mathbf{ex}(G)$ denote the *excess* of G, that is the minimum number of edges that must be removed from G to ensure that every connected component is acyclic or unicyclic.

Analysis is based on the following critical observation (which is proven, for instance, as Lemma 5 of [ADW14]).

Theorem 2. *Let G be the graph representation of a Cuckoo hash table with a stash of size s. Then the build succeeds if and only if $\boldsymbol{ex}(G) \leq s$.*

6 Kirsch et al: Cuckoo Hashing with a Constant Sized Stash

Here we present the analysis of Kirsch et al [KMW09]. While they delt only with the case where the stash is constant, their analysis is the foundation of Goodrich and Mitzenmacher's analysis of the case with super-constant stashes and the tighter analysis presented later in this paper.

As explained in Section 5, the analysis first treats the problem as looking at the distribution of the excess of the corresponding graph distribution.

The first step is to describe distributions of these graphs. $G(m, m, D)$ denotes the distribution of graphs generated by picking a bipartite graph with m nodes in each part, picking a number of edges according to D, and assigning each edge a left-endpoint chosen uniformally at random from one part and a right-endpoint chosen uniformally at random from the other part. $G(m,m,n)$ describes the actual distribution of Cuckoo graphs, where n, in slight abuse of notation, also represents a probability distribution entirely concentrated at the value n.

Kirsch et al. propose to instead look at the distribution $G(m,m,Po(\lambda))$, where $Po(\lambda)$ represents the Poisson distribution with parameter λ. This has the desirable property that the multiplicity of each edge is distributed according to $Po(\frac{\lambda}{m^2})$ and is independent.

They then show that for $n(1+\epsilon_0) \leq \lambda \leq m(1-\epsilon_0)$ for some constant ϵ_0, the probability that $Po(\lambda) < n$ is negligible. Specifically: $\mathbf{Pr}(Po(\lambda) < n) \leq e^{-\lambda}\left(\frac{e\lambda}{n}\right)^n \leq e^{-n(\epsilon_0 - \ln(1+\epsilon_0))} \leq e^{-\Omega(n)}$. Since $n = \omega(\log(N))$ this probability is negligible in N.

Let $G_0 \leftarrow G(m,m,X)$ where X is distributed according to $Po(\lambda)$ conditioned on $X \geq n$. They showed that any upper bound on $\mathbf{ex}(G_0)$ will also apply to $\mathbf{ex}(G(m,m,n))$. This is evident since G_0 can be viewed as first picking a graph from $G(m,m,n)$ and then adding a further $X - n$ edges. Doing this will never reduce the excess, and may increase it.[3]

Following this, Kirsch et al present the following important result:

Lemma 1. *Lemma 2.7 of [KMW09]. There exists some constant,* $0 < \beta < 1$ *such that for any fixed vertex* v *and integer* $k \geq 0$,

$$\boldsymbol{Pr}(|C_v| \geq k) \leq \beta^k$$

Here C_v is the connected component containing vertex v and $|C_v|$ is the number of edges it contains.

They then examine the cyclotomic number of each component, $\gamma(C_v)$. First they calculate this conditioned on the number of edges in the component:

Lemma 2. *Lemma 2.8 of [KMW09]. For every vertex* v *and* $t, k, n \geq 1$

$$\boldsymbol{Pr}(\gamma(C_v) \geq t \,|\, |C_v| = k) \leq \left(\frac{3e^5k^3}{m}\right)^t$$

Combining these yields that for any constant t:

$$\mathbf{Pr}(\gamma(C_v) \geq t) \leq \sum_{k=1}^{\infty} \left(\frac{3e^5k^3}{m}\right)^t \beta^k$$

Using the assumption that t is a constant, this simplifies to $\mathcal{O}(n^{-t})$. (As a result of assuming constant t, their later equations will only apply for constant-sized stashes.)

It follows naturally that the *excess* of a component has the following bound for any constant s:

$$\mathbf{Pr}(\mathbf{ex}(C_v) \geq s) = \mathbf{Pr}(\gamma(C_v) \geq s+1) \leq \mathcal{O}(n^{-(s+1)})$$

[3] Kirsch et al prove a more general statement using the language of stochastic dominance. However, this generality is not needed for their main result.

The authors then observe that, even though the distributions $\mathbf{ex}(C_v)$ are not independent across different components, they are in fact *negatively correlated.* Furthermore, the number of vertices is fewer than the number of components. Therefore, they show that the sum of $2m$ independent samples of $\mathbf{ex}(C_v)$, will stochastically dominate the actual excess of the graph. If $B_1, \ldots, B_{2m}$ are $2m$ independent samples from $\gamma(C_v)$, then:

$$\mathbf{Pr}(\mathbf{ex}(G) \geq s) \leq \mathbf{Pr}\left(\sum_{i=1}^{2m} B_i \geq s + |i : B_i \geq 1|\right)$$

By separating the sum according to the number of components with non-zero excess and assuming that s is a constant, the authors derive the desired bound of $\mathcal{O}(n^{-s})$.

In short, Kirsch et al. provided the first analsyis of Cuckoo Hashing with a stash, and present a useful framework for analysis by examining the excess of random components in a related Poisson-based graph representation. However, their result assumes the stash size is constant and does not make the constants explicit.

7 Goodrich and Mitzenmacher: Privacy-Preserving Access of Outsourced Data via Oblivious RAM Simulation

Goodrich and Mitzenmacher constructed an Oblivious RAM scheme using Cuckoo hashing. Firstly, this required that the failure probability be *negligible*, which could not be satisfied by the previous analysis of Kirsch et al [KMW09] which only provided failure probability $\mathcal{O}(n^{-s})$ for constant-sized s. Furthermore, the ORAM application now required failure to be negligible in the size of the ORAM, N, not in the size of the table, n, where n could by poly-logarithmic in N.

This therefore required a much closer analysis. This analysis is presented in Appendix C of their paper, but despite its obscure location, its result has been very important for later ORAM work and referenced extensively [KLO12, GMOT12, LO13]. In particular they show that for $n = \Omega(\log^7(N))$, and $s = \Theta(\log(N))$, the failure probability is negilible in N. Considering how cited this result is, the analysis is slightly cursory, so below we fill in some of the missing details.

They start from the results of Kirsch et al. This implicitly re-uses the Poissonization argument, which holds since $n = \omega(\log(N))$ is still satisfied. Specifically, they begin with the results that

$$\mathbf{Pr}(\gamma(C_v) \geq t||C_v| = k) \leq \left(\frac{3e^5k^3}{m}\right)^t$$

and that for some constant $0 \leq \beta \leq 1$

$$\mathbf{Pr}(|C_v| = k) \leq \beta^k$$

Combining these yields that

$$\begin{aligned}\mathbf{Pr}(\gamma(C_v) \geq t) &\leq \sum_{k=1}^{\infty} \mathbf{Pr}(\gamma(C_v)||C_v| = k)\mathbf{Pr}(|C| \geq k) \\ &\leq \sum_{k=1}^{\infty} \min\left(\left(\frac{3e^5k^3}{m}\right)^t, 1\right)\beta^k\end{aligned}$$

β^k is a geometric sequence in k. Recalling that $0 < \beta < 1$, this means that $\sum_{i=a}^{\infty} \beta^k = \frac{\beta^a}{1-\beta}$. They observe that when $k = \Omega(\log^2 N)$, β^k will be negligible in N, so

$$\sum_{k=\log^2(N)}^{\infty} \min\left(\left(\frac{3e^5k^3}{m}\right)^t, 1\right)\beta^k = N^{-\omega(1)}$$

This means the expression will be negligible if the sum of the terms from $k = 1$ to $k = \mathcal{O}(\log^2(N))$ is negligible.

They assume that $m = \Theta(n) = \Omega(\log^7 N)$. They argue that in this case $\mathbf{Pr}(\gamma(C_v) \geq j + 1)$ is at most $m^{-1-\alpha j}$ for some constant α. A proof of this is missing in the paper, but it can be shown that $\mathbf{Pr}(\gamma(C_v) \geq j+1) = \mathcal{O}(m^{-1-\alpha j})$ by the following argument. Given that $k = \mathcal{O}(\log^2(N))$,

$$\frac{3e^5k^3}{m} = \mathcal{O}\left(\frac{1}{\log(N)}\right) = \mathcal{O}\left(m^{-\frac{1}{7}}\right)$$

Therefore,

$$\sum_{k=1}^{\log^2(N)} \min\left(\left(\frac{3e^5k^3}{m}\right)^t, 1\right)\beta^k \leq \sum_{k=1}^{\log^2(N)} \left(3e^5\right)^t m^{-\frac{t}{7}}$$

Observe that, from the constant-stash case, $\mathbf{Pr}(\gamma(C_v) \geq t+1) = \mathcal{O}(m^{-1-t})$. For $t \geq 9$ and sufficiently large m,

$$\begin{aligned}\mathbf{Pr}(\gamma(C_v) \geq t + 1) &\leq \text{neg}(N) + \sum_{k=1}^{\log^2(N)} \left(3e^5\right)^{t+1} m^{-\frac{t+1}{7}} \\ &\leq \text{neg}(N) + \log^2(N)\left(3e^5\right)^{t+1} m^{-\frac{9}{7}-\frac{t-8}{7}} \\ &\leq \text{neg}(N) + m^{\frac{2}{7}}\left(3e^5\right)^{t+1} m^{-\frac{9}{7}-\frac{t-8}{7}} \\ &\leq \text{neg}(N) + \left(3e^5\right)^{t+1} m^{-1-\frac{t}{63}} \\ &\leq \mathcal{O}(m^{-1-\alpha t}) \text{ for some constant } \alpha\end{aligned}$$

Since $\mathbf{Pr}(\gamma(C_v) \geq t + 1) \leq \mathcal{O}(m^{-1-t})$ for $1 \leq t \leq 8$ and (with a different constant in the $\mathcal{O}$-notation) $\mathbf{Pr}(\gamma(C_v) \geq t + 1) \leq \mathcal{O}(m^{-1-\alpha t})$ for $t \geq 9$, $\mathbf{Pr}(\gamma(C_v) \geq t + 1) \leq cm^{-1-\alpha t}$ for some constants α and c.

Continuing from their claim that $\mathbf{Pr}(\gamma(C_v) \geq t+1) \leq m^{-1-\alpha t}$ they continue their analysis based on that of Theorem 2.2 of [KMW09]. We present the analaysis below, but include the constant c.

$$\begin{aligned}
\mathbf{Pr}(\mathbf{ex}(G) \geq s) &\leq \sum_{\substack{j_1,\ldots,j_{2m}\\ \sum_{i=1}^{2m} j_i = s}} \prod_{\substack{i=1,\ldots,2m\\ j_i \geq 1}} cm^{-1-\alpha j_i}\\
&\leq \sum_{k=1}^{2m} \sum_{\substack{j_1,\ldots,j_{2m}\\ \sum_{i=1}^{2m} j_i = s\\ |\{i: j_i \geq 1\}| = k}} c^k m^{-\alpha s - k}\\
&\leq \sum_{k=1}^{2m} \binom{2m}{k} s^k c^k m^{-\alpha s - k}\\
&\leq \sum_{k=1}^{2m} \left(\frac{2me}{k}\right)^k s^k c^k m^{-\alpha s - k}\\
&\leq m^{-\alpha s} \sum_{k=1}^{2m} \left(\frac{2esc}{k}\right)^k\\
&\leq m^{-\alpha s} \sum_{k=1}^{2m} e^{2sc}\\
&\leq m^{-\alpha s} 2me^{2sc}\\
&\leq m^{1+\log_m(2)-(\alpha - 2c\log_m(e))s}\\
&\leq m^{-\Omega(s)}
\end{aligned}$$

For $s = \Theta(\log(N))$ and $m = \Omega(\log^7(N))$, this is negligible in N.

8 Aumüller et al. Explicit and Efficient Hash Families Suffice for Cuckoo Hashing with as Stash

Aumüller et al. [ADW14] present an elegant alternative analysis of Cuckoo Hashing with a stash. This analysis also treats the problem by representing it as a problem on random graphs and determining whether the excess of the graph is above a certain threshold. However, unlike the approach of Kirsch et al [KMW09], they do this by counting the number of possible graphs that would result in a stash overflow, and then determining the probability that the graph representation of the Cuckoo Hash table is such a graph.

They define $N(t, \ell, \gamma, \zeta)$ to be the number of non-isomorphic multi-graphs with t edges, ℓ leaf edges, cyclotomic number γ and ζ components. They prove (Lemma 4) using simple inductive arguments, that $N(t, \ell, \gamma, \zeta) = t^{O(\ell+\gamma+\zeta)}$.

They then define (Definition 6) an *excess-$(s+1)$ core graph* to be a leafless graph of excess exactly $s+1$ in which every component contains at least 2 cycles. An excess-$(s+1)$ core graph can be thought of as a minimal version of a graph that still contains excess $s+1$, and they prove that any graph that has excess at least $s+1$ will contain an excess-$(s+1)$ core graph as a subgraph. Hence, the probability that $\mathbf{ex}(G) \geq s+1$ is the probability that G contains a sub-graph that is an excess-$(s+1)$ core graph.

Let G' be an excess-$(s+1)$ core graph, with connected components C_i for $1 \leq i \leq \zeta$. By definition an excess-$(s+1)$ core graph will have each connected component contribute at least one to the excess, so $\zeta \leq s+1$. Furthermore, if $\gamma_i = \gamma(C_i)$, then $(s+1) = \sum_{i=1}^{\zeta}(\gamma_i - 1) = \gamma - \zeta$, so $\gamma \leq 2(s+1)$. Lastly, an excess-$(s+1)$ core graph by definition has no leaves. This means that the number of excess-$(s+1)$ core graphs with t edges is $N(t, \ell, \gamma, \zeta) = t^{O(\gamma+\zeta)} = t^{O(s)}$.

They then proceed to determine the number of *labelled* excess-$(s+1)$ core graphs that are sub-graphs of bipartite graph. First it is necessary to observe that the number of vertices is $t - \gamma + \zeta = t - (s+1)$. Since the graph is bipartite, each connected component of the sub-graph is also bipartite, so the part-assignment of a single vertex in that component determines the part-assignment of all other vertices. There are therefore $2^{\zeta} \leq 2^{s+1}$ possible choices of assignments of vertices to parts. Given the assignment of parts for each vertex, there are at most $m^{t-(s+1)}$ assignments of labels. Given a set of t edges, the number of ways the edges can be labelled is $t!$. This shows that the total number of possible labelled excess-$(s+1)$ core graphs (given a set of t edge labels) is upper-bounded by:

$$t!2^{s+1}m^{t-s-1}t^{O(s)}$$

Given n edges, there are $\binom{n}{t}$ subsets of these edges. There are therefore at most $\binom{n}{t}t!2^{s+1}m^{t-s-1}t^{O(s)}$ labelled subgraphs with t edges that form an excess-$(s+1)$ core graph.

Each edge is chosen from $[m]^2$ (since there are m vertices in each part). Therefore, the probability that there is a labelling that corresponds to an excess-$(s+1)$ core graph with t edges is at most

$$\binom{n}{t}\frac{t!2^{s+1}m^{t-s-1}t^{O(s)}}{m^{2t}} \leq \frac{2^{s+1}}{m^{s+1}}\frac{n^t t^{O(s)}}{m^t} = \frac{2^{s+1}}{m^{s+1}}\frac{t^{O(s)}}{(1+\epsilon)^t}$$

An excess-$(s+1)$ core graph, must have at least $s+3$ edges, so if we sum over all $s+3 \leq t \leq n$ we obtain the final result:

$$\mathbf{Pr}(\mathbf{ex}(G) \geq s+1) \leq \frac{2^{s+1}}{n^{s+1}} \sum_{s+3 \leq t \leq n} \frac{t^{O(s)}}{(1+\epsilon)^t} \tag{1}$$

For some constant s, this yields the following failure probability (Lemma 7 of [ADW14]):

$$\mathbf{Pr}(\mathbf{ex}(G) \geq s+1) \leq \mathcal{O}\left(\frac{1}{n^{s+1}}\right)$$

They also demonstrates bounds for super-constant s. Specifically, they show that for sufficiently large n, and for a certain, super-constant, range of s, the following bound holds (Theorem 2 of [ADW14]):

$$\mathbf{Pr}(\mathbf{ex}(G) \geq s+1) \leq \mathcal{O}\left(\frac{1}{n^{\frac{s}{2}}}\right)$$

However, the analysis of Aumüller et al does not provide good concrete bounds as is. In the proof of their analysis of a super-constant sized stash (Theorem 2), they state that an initial examination of their proofs shows yields a constant of 27 in the big-oh notation in the exponent in Equation 1. Furthermore, this requires the table size to be a large polynomial in the size of s, specifically $n \geq s^{3*27} = s^{81}$. In the case where $N >> n$ and failure negligble in N is required, this requires that $n \geq \log^{81}(N)$, which is much looser than the bound $n \geq \log^{7}(N)$ of [GM11]. It would be an interesting further work to determine if a more careful analysis on the number of excess-$(s+1)$ core graphs could yield bounds that are useful for computing required stash sizes for concrete failure probabilities.

In addition to this analysis, Aumüller et al present explicit efficient hash-functions. They show that even though these hash functions no longer have outputs that are chosen uniformly at random and independent, these hash functions are still sufficiently random that the failure probability is not significantly increased.

9 A Tight Analysis

In this section we present an analysis of Cuckoo Hashing with a stash. This analysis shows that for $s = \Theta(\log(N))$ the failure probability is negligible for any $n = \omega(\log(N))$.

As with Kirsch et al. we look at graphs chosen from $G(m, m, Po(\lambda))$ to upper-bound those chosen from $G(m, m, n)$. This means that the multiplicity of each edge is chosen from $Po\left(\frac{\lambda}{m^2}\right)$, where $n(1+\epsilon_0) \leq \lambda \leq m(1-\epsilon_0)$. This adds a failure probability of $e^{-n(\epsilon_0 - \ln(1+\epsilon_0))}$ to account for the possibility that the total number of edges is fewer than n.

We now bound the cyclotomic number of connected components in $G(m, m, Po(\lambda))$ given their size where a component's size is defined as the number of vertices in the component.[4]

Theorem 3. *Given a random vertex v in $G(m, m, Po(\lambda))$, and letting C_v be the connected component containing v*

$$\boldsymbol{Pr}(\gamma(C_v) \geq t || C_v| = k) \leq \left(\frac{ek^2(1-\epsilon_0)}{4mt}\right)^t$$

[4] Note that this differs from [KMW09], in which the size of a component is the number of edges it contains.

Proof. Imagine a Breadth First Search on a component of this graph. We can execute the Breadth First Search such that if a vertex at depth d has multiple edges to vertices at depth $d-1$ we only observe one of these edges. Once the BFS is complete, we can then observe the number of edges that were not observed during the BFS.

Let us make some observations between the edges that could have not been observed in the graph. If there is some vertex u at level d, there cannot be an unobserved edge between it and any vertex w at level $d' >= d+2$, as if such an edge existed, w would have been found earlier during the BFS and placed at level $d+1$. Recall also that the graph is bipartite. We can therefore see that all vertices at a given level are in the same part. This can be shown by induction: it is true trivially of the first level which contains only a single vertex. Given it is true of a certain level d, it is true of level $d+1$ since all vertices in level $d+1$ are neighbors of vertices in level d so must all be in the other part of the graph as those in level d, so are all in the same part of the graph as each other. It follows that there cannot be any edges between vertices at the same level. Let a be the number of vertices in the connected component C_v, where C_v has k vertices total. The number of unobserved vertex pairs that may have edges between them is therefore at most $a(k-a)$. This is maximized by $a = \frac{k}{2}$ and results in there being $\frac{k}{4}$ pairs of vertices in C_v that may have vertices between them.

Of these $k-1$ are known to have at least one edge between them, these being the edges that were found during the BFS. Since the Poisson distribution is exponentially decreasing, the number of *additional* edges between each of these pairs of vertices, is stochastically dominated by $Po\left(\frac{\lambda}{m^2}\right)$. The number of edges between pairs of vertices such that the multiplicity of the edges between them was not observed is exactly $Po\left(\frac{\lambda}{m^2}\right)$ and there are at most $\frac{k^2}{4}-(k-1)$ such edge pairs. Therefore, the total number of unobserved edges is stochastically dominated by $Po\left(\frac{k^2\lambda}{4m^2}\right)$.

Given a standard bound of the Poisson distribution, for any $t \geq 1$ this means that

$$\begin{aligned}\mathbf{Pr}\left(\gamma(C_v) \geq t ||C_v| = k\right) &\leq \left(\frac{ek^2(1-\epsilon_0)}{4mt}\right)^t e^{-\frac{k^2(1-\epsilon_0)}{4m}} \\ &\leq \left(\frac{ek^2(1-\epsilon_0)}{4mt}\right)^t\end{aligned}$$

□

Now we need to upper-bound the probability that $|C_v| = k$. Kirsch et al show that $\mathbf{Pr}(|C_v| \geq k) \leq \beta^k$ for a constant $\beta < 1$. We will prove a slightly different bound that will be more useful for our later analysis. It also appears in their analysis, they switch back to analyzing $G(m,m,n)$, whereas we will continue to analyze $G(m,m,Po(\lambda))$.

Theorem 4. *Given a random vertex v in $G(m, m, Po((1-\epsilon_0)m))$, and letting C_v be the connected component containing v, where ϵ_0 is a constant in $(0,1)$ and $\epsilon_1 = \frac{\epsilon_0^2}{2-\epsilon_0}$, then*

$$\boldsymbol{Pr}(|C_v| = k) \leq \frac{2}{k(1-\epsilon_0)} e^{-\epsilon_1 k}$$

Proof. Let us return to the BFS algorithm. Given that the node u has been found in the BFS, and that node w has not yet been found, the probability that w is a child of u is:

$$\begin{aligned}\mathbf{Pr}\left(Po\left(\frac{\lambda}{m^2} \geq 1\right)\right) &= 1 - \mathbf{Pr}\left(Po\left(\frac{\lambda}{m^2}\right) = 0\right) \\ &= 1 - e^{-\frac{\lambda}{m^2}} \\ &\leq 1 - \left(1 - \frac{\lambda}{m^2}\right) \\ &\leq \frac{\lambda}{m^2}\end{aligned}$$

Now, some of the nodes will have already been observed by the BFS. These have zero probability of being a child of u. For nodes that have not been observed by the BFS, the probabilities that they are a child of u is independent. Therefore, the number of children of u is stochastically dominated by $\text{Bin}(m, \frac{\lambda}{m^2})$. Let $r = \frac{\lambda}{m} = 1 - \epsilon_0$. The number of children is therefore stochastically dominated by $\text{Bin}(m, \frac{r}{m})$.

From Pitman [Pit98] we have the result that if a branching process has children chosen independently from the distribution X_i, the probability that the total progeny is k is (exactly) $\frac{1}{k}\mathbf{Pr}(S_k = k-1)$ where S_k is the sum of k independent samples from X_i. In this case, we instead have that the number of children is stochastically dominated by $\text{Bin}(m, \frac{\lambda}{m^2})$, therefore:

$$
\begin{aligned}
\mathbf{Pr}(|C_v| = k) &\leq \frac{1}{k}\mathbf{Pr}(\mathrm{Bin}(mk, \frac{r}{m}) = k-1) \\
&\leq \frac{1}{k}\left(\frac{r}{m}\right)^{k-1}\left(1-\frac{r}{m}\right)^{mk-(k-1)}\binom{mk}{k-1} \\
&\leq \left(\frac{r}{m}\right)^{-1}\left(\frac{r}{m}\right)^{k}\left(1-\frac{r}{m}\right)\left(1-\frac{r}{m}\right)^{mk-k}\frac{1}{mk-(k-1)}\frac{(mk)!}{k!(mk-k)!} \\
&\leq \frac{1}{mk-(k-1)}\frac{m}{r}\left(1-\frac{r}{m}\right)\mathbf{Pr}(\mathrm{Bin}(mk, \frac{r}{m}) = k) \\
&\leq \frac{1}{mk-(k-1)}\left(\frac{m}{r}-1\right)\mathbf{Pr}(\mathrm{Bin}(mk, \frac{r}{m}) = k) \\
&\leq \frac{m}{mk-(k-1)}\left(\frac{1}{r}\right)\mathbf{Pr}(\mathrm{Bin}(mk, \frac{r}{m}) = k) \\
&\leq \frac{1}{k-\frac{k-1}{m}}\left(\frac{1}{r}\right)\mathbf{Pr}(\mathrm{Bin}(mk, \frac{r}{m}) = k) \\
&\leq \frac{1}{k-\frac{k}{2}}\left(\frac{1}{r}\right)\mathbf{Pr}(\mathrm{Bin}(mk, \frac{r}{m}) = k) \\
&\leq \frac{2}{k}\left(\frac{1}{r}\right)\mathbf{Pr}(\mathrm{Bin}(mk, \frac{r}{m}) = k) \\
&\leq \frac{2}{kr}\mathbf{Pr}(\mathrm{Bin}(mk, \frac{r}{m}) \geq k)
\end{aligned}
$$

The expected value of $\mathrm{Bin}(mk, \frac{r}{m})$ is rk. Setting $s = \frac{1}{r} - 1 = \frac{1-r}{r} = \frac{\epsilon_0}{1-\epsilon_0} > 0$ and observing $k = (1+s)rk$ Standard Chernoff bounds show that

$$
\begin{aligned}
\mathbf{Pr}(\mathrm{Bin}(mk, \frac{r}{m}) \geq k) &\leq e^{-\frac{s^2 rk}{2+s}} \\
&\leq e^{-\frac{\left(\frac{\epsilon_0}{1-\epsilon_0}\right)^2 (1-\epsilon_0)k}{2+\frac{\epsilon_0}{1-\epsilon_0}}} \\
&\leq e^{-\frac{\epsilon_0^2 k}{2(1-\epsilon_0)+\epsilon_0}} \\
&\leq e^{-\frac{\epsilon_0^2 k}{2-\epsilon_0}} \\
&\leq e^{-\epsilon_1 k}
\end{aligned}
$$

where $\epsilon_1 = \frac{\epsilon_0^2}{2-\epsilon_0}$.

Therefore $\mathbf{Pr}(|C_v| = k) \leq \frac{2}{kr}e^{-\epsilon_1 k}$. □

Combining Theorems 3 and 4 yields:

$$\begin{aligned}
\mathbf{Pr}(\gamma(C_v) \geq t) &\leq \sum_{k=1}^{2m} \left(\frac{ek^2r}{4mt}\right)^t \frac{2}{kr} e^{-\epsilon_1 k} \\
&\leq \frac{2}{r}\left(\frac{er}{4mt}\right)^t \sum_{k=1}^{2m} k^{2t-1} e^{-\epsilon_1 k} \qquad (2)
\end{aligned}$$

We will need the following Lemma to simplify this further:

Lemma 3. *Let $t \geq 1$, $\epsilon_1 \in (0,1)$. Then:*

$$\sum_{k=1}^{2m} k^{2t-1} e^{-\epsilon_1 k} \leq 2e\left(\frac{2t}{\epsilon_1 e}\right)^{2t}$$

Proof. It is possible to approximate a summation with an integral, using the same methods as Reinman sums but in reverse. Let $f(x)$ be a continuous function that is monotonically increasing until a maximum point x_{max}, after which x is monotonically decreasing. Let $x' = \lfloor x_{max} \rfloor$. Let $h(x) = \min(f(\lfloor x \rfloor), f(\lfloor x \rfloor + 1))$. Let us observe how $\sum_{x=a}^{b} h(x)$ approximates $\int_a^{b+1} f(x)dx$.

Observe that for any integer a, $h(x)$ is the same for all $x \in [a, a+1)$. Since $f(x)$ has no local minima and is continuous, the minimum value of $f(x)$ over the range $[a, a+1)$ is either at $f(a)$ or $f(a+1)$. Therefore $f(x) \geq \min(f(a), f(a+1)) = h(x)$ for $x \in [a, a+1)$. Since this applies to the interval $[a, a+1)$ for any integer a, $h(x) \leq f(x)$ for all x. Hence, for any integers a and b, $\sum_a^b h(x) = \int_a^{b+1} h(x) \leq \int_a^{b+1} f(x)dx$.

$$\begin{aligned}
\int_a^{b+1} f(x)dx &\geq \sum_a^b h(x) \\
&\geq \sum_a^{x'-1} f(x) + min(f(x'), f(x'+1)) + \sum_{x'+1}^{b} f(x+1) \\
&\geq \left(\sum_a^{b+1} f(x)\right) - max(f(x'), f(x'+1) \\
&\geq \left(\sum_a^{b+1} f(x)\right) - f(x_{max})
\end{aligned}$$

Hence:

$$\sum_a^b f(x) \leq \int_a^b f(x)dx + f(x_{max})$$

Let $f(x) = x^{2t-1}e^{-\epsilon_1 x}$ where $t \geq 1$ and $0 < \epsilon_1 < 1$. First we need to show that it is a function that is monitonically increasing, then monitonically decreasing.

$$f'(x) = (2t-1)x^{2t-2}e^{-\epsilon_1 x} - \epsilon_1 x^{2t-1}e^{-\epsilon_1} \qquad = x^{2t-2}e^{-\epsilon_1}(2t-1-\epsilon_1 x)$$

Observe that x^{2t-2} and $e^{-\epsilon_1}$ are both positive. Therefore $f'(x)$ will be positive when $x < \frac{2t-1}{\epsilon_1}$, $f'(x) = 0$ at $x = \frac{2t-1}{\epsilon_1}$ and will be negative when $x > \frac{2t-1}{\epsilon_1}$. Therefore this function is monotonically increasing, then monitonically decreasing, as required, with $x_{max} = \frac{2t-1}{\epsilon_1}$. We can easily calculate:

$$\begin{aligned} f(x_{max}) &= \left(\frac{2t-1}{\epsilon_1}\right)^{2t-1} e^{-(2t-1)} \\ &= \left(\frac{2t-1}{\epsilon_1 e}\right)^{2t-1} \end{aligned}$$

Hence the inequality applies to the sum and

$$\begin{aligned} \sum_{k=1}^{2m} k^{2t-1}e^{-\epsilon_1 k} &\leq \int_1^{2m} x^{2t-1}e^{-\epsilon_1 x}dx + \left(\frac{2t-1}{\epsilon_1 e}\right)^{2t-1} \\ &\leq \int_0^{\infty} x^{2t-1}e^{-\epsilon_1 x}dx + \left(\frac{2t-1}{\epsilon_1 e}\right)^{2t-1} \end{aligned}$$

By a standard integral identity, $\int_0^\infty x^{2t-1}e^{-\epsilon_1 x}dx = \frac{(2t-1)!}{\epsilon_1^{2t}}$. Furthermore, a standard factorial approximation shows that $(2t-1)! \leq (2t)^{2t}e^{-2t-1}$. Hence

$$\begin{aligned} \sum_{k=1}^{2m} k^{2t-1}e^{-\epsilon_1 k} &\leq \left(\frac{2t}{\epsilon_1}\right)^{2t} e^{-(2t-1)} + \left(\frac{2t-1}{\epsilon_1 e}\right)^{2t-1} \\ &\leq \left(\frac{2t}{\epsilon_1}\right)^{2t} e^{-(2t-1)} + \left(\frac{2t}{\epsilon_1}\right)^{2t-1} e^{-(2t-1)} \\ &\leq 2\left(\frac{2t}{\epsilon_1}\right)^{2t} e^{-(2t-1)} \\ &\leq 2e\left(\frac{2t}{\epsilon_1 e}\right)^{2t} \end{aligned}$$

where we use the fact $0 < \epsilon_1 < 1$.

□

Applying Lemma 3 to equation 2 yields the following, fairly concise result:

$$\begin{aligned}\mathbf{Pr}(\gamma(C_v) \geq t) &\leq \frac{2}{r}\left(\frac{er}{4mt}\right)^t 2e\left(\frac{2t}{\epsilon_1 e}\right)^{2t} \\ &\leq \frac{4e}{r}\left(\frac{rt}{me\epsilon_1^2}\right)^t\end{aligned}$$

The following corrolary immediately follows:

Corollary 2. *Let constants $\epsilon_2 = \frac{4e}{1-\epsilon_0}$ and $\epsilon_3 = \frac{1-\epsilon_0}{e\epsilon_1^2}$, where $\epsilon_1 = \frac{\epsilon_0^2}{2-\epsilon_0}$. Then the cyclotomic number and excess of the connected component of a random vertex in $G(m, m, Po\left((1-\epsilon_0)m\right))$ are upper-bounded as follows:*

$$\boldsymbol{Pr}(\gamma(C_v) \geq t) \leq \epsilon_2\left(\frac{\epsilon_3 t}{m}\right)^t$$

$$\boldsymbol{Pr}(\boldsymbol{ex}(C_v) \geq s) \leq \epsilon_2\left(\frac{\epsilon_3(s+1)}{m}\right)^{s+1}$$

This bounds the excess of a single component. We would like to use this to bound the excess of the entire graph. However, the excess of different components is not independent. Thankfully, as Kirsch et al observed, $\gamma(C_v)$ are negatively correlated across different components, and are therefore stochastically dominated by the sum of independent samples.

Kirsch et al. formalized this as the following lemma, which they prove in their paper.

Lemma 4. *(Lemma 2.10 of [KMW09]) Fix some ordering $v_1, \ldots, v_{2m}$ of the vertices. For $i = 1, \ldots 2m$, let $C'_{v_i} = C_{v_i}$ if v_i is the first vertex in the ordering to appear in C_v, and let C'_{v_i} be the empty graph on the $2m$ vertices otherwise. Let $C''_{v_1}, \ldots, C''_{v_{2m}}$ be [5] independent random variables such that each C''_{v_i} is distributed as C_{v_i}. Then $(C''_{v_1}, \ldots, C''_{v_{2m}})$ stochastically dominates $(C'_{v_1}, \ldots, C'_{v_{2m}})$.*

In brief, Kirsch et al prove this by showing that, if the C_{v_i} are sampled in order, then C'_{v_i} will either be sampled from the empty graph (if v_i was already found in a component) or will be sampled as with C_{v_i} but with all previously-found vertices removed from the graph.

It therefore follows that $\mathbf{ex}(G) = \sum_{i=1}^{2m} \mathbf{ex}(C'_{v_i}) \leq \sum_{i=1}^{2m} \mathbf{ex}(C''_{v_i})$

We make the following additional observation. Recall that the graph is bipartite and each part contains m vertices. We will pick our indexing such that the vertices 1 through m are all in a single part. Since every component that contains a cycle must contain vertices in both parts of the graph, then the vertices indexed $m+1$ to $2m$ will, if they are in a component that contains a

[5] The Lemma in Kirsch et al. states this as up to C''_{v_m} but $2m$ such variables are needed for their lemma.

cycle, not be the lowest index in that component. Therefore $\mathbf{ex}(C'_i) = 0$ for $m+1 \le i \le 2m$.

It also follows from Lemma 2.10 of [KMW09] that $(C''_{v_1}, \ldots, C''_{v_m})$ stochastically dominates $(C'_{v_1}, \ldots, C'_{v_m})$ Hence

$$\mathbf{ex}(G) = \sum_{i=1}^{m} \mathbf{ex}(C'_{v_i}) \le \sum_{i=1}^{m} \mathbf{ex}(C''_{v_i})$$

To complete our analysis, we will need the following Fact. It is simple to state, but slightly tedius to prove, but is worth to work.

Fact 1. *For all integers $s \ge 2$, $\sum_{a=1}^{s-1}(a+1)^{a+1}(s+1-a)^{s+1-a} \le \epsilon_4(s+1)^{s+1}$*

Proof. By calculation, this is true for $s \in \{2,3,4,5,6,7,8\}$, for which the left-hand side values are, respectively $\{16, 216, 2777, 38824, 607534, 10707768, 212342547\}$ and the right-hand side values are respectively $\{24.03,\ 227.84,\ 2781.25,\ 41523.84,\ 732953.27,\ 14931722.24,\ 344804235.2\}$. For $s > 8$ we prove by induction.

Given that it holds true for $s \le 8$, let us show it holds true for $s+1$.

$$\begin{aligned}
&\sum_{a=1}^{s}(a+1)^{a+1}(s+2-a)^{s+2-a} \\
&\le \sum_{a=1}^{t}(a+1)^{a+1}(s+2-a)^{s+2-a} + \sum_{a=t+1}^{s-1}(a+1)^{a+1}(s+2-a)^{s+2-a} + (s+1)^{s+1}2^2 \\
&\le \sum_{a=1}^{t}(a+1)^{a+1}(s+2-a)^{s+2-a} + (s+1-t)e\sum_{a=t+1}^{s-1}(a+1)^{a+1}(s+1-a)^{s+1-a} + (s+1)^{s+1}2^2 \\
&\le \sum_{a=1}^{t}(a+1)^{a+1}(s+2-a)^{s+2-a} + (s+1-t)e \\
&\qquad \left(\sum_{a=1}^{s-1}(a+1)^{a+1}(s+1-a)^{s+1-a} - \sum_{a=1}^{t}(a+1)^{a+1}(s-a+1)^{s-a+1}\right) + (s+1)^{s+1}2^2 \\
&\le e\sum_{a=1}^{t}(a+1)^{a+1}(s+2-a)(s+1-a)^{s+1-a} + (s+1-t)e\epsilon_4(s+1)^{s+1} \\
&\qquad - e(s+1-t)\sum_{a=1}^{t}(a+1)^{a+1}(s-a+1)^{s-a+1} + (s+1)^{s+1}2^2 \\
&\le e\sum_{a=1}^{t}(a+1)^{a+1}(s-a+1)^{s-a+1}((s+2-a)-(s+1-t)) + \epsilon_4(s+2)^{s+2} \\
&\qquad - te\epsilon_4(s+1)^{s+1} + (s+1)^{s+1}2^2 \\
&\le \epsilon_4(s+2)^{s+2} + 2^2(s+1)^{s+1} + e\sum_{a=1}^{t}(a+1)^{a+1}(s-a+1)^{s-a+1}(t+1-a) - te\epsilon_4(s+1)^{s+1}
\end{aligned}$$

Setting $t = 3$ yields:

$$\leq \epsilon_4(s+2)^{s+2} + 2^2(s+1)^{s+1} + e2^2s^s3 + e3^3(s-1)^{s-1}2 + e4^4(s-2)^{s-2} - 3e\epsilon_4(s+1)^{s+1}$$
$$\leq \epsilon_4(s+2)^{s+2} + (s+1)^{s+1}\left(2^2 + \frac{12}{s} + \frac{54}{es(s-1)} + \frac{256}{e^2s(s-1)(s-2)} - 3e\epsilon_4\right)$$

For $s \geq 8$, the term $2^2 + \frac{12}{s} + \frac{54}{es(s-1)} + \frac{256}{e^2s(s-1)(s-2)} \leq 5.5$. Since $3e\epsilon_4 > 5.5$ the inequality simplifies to:

$$\sum_{a=1}^{s}(a+1)^{a+1}(s+2-a)^{s+2-a} \leq \epsilon_4(s+2)^{s+2}$$

Since it holds true up to $s = 2, \ldots, 8$ by inspection, and holds true for $s \geq 8$ by induction, the statement is true for all $s \geq 2$. □

Lemma 5. *Let $U(s,q)$ be the set of sequences of positive integers, where $T \in U$ if and only if $|T| = q$ and $\sum_{1\leq i\leq q} T_i = s$, where $s \geq q \geq 1$. Then $\sum_{T\in U(s,q)} \prod_{1\leq i\leq q}(T_i+1)^{T_i+1} \leq 0.89^{q-1}(s+1)^{s+1}$*

Proof. We proceed by induction on the length of the sequences. For $q = 1$, U contains a single sequence T with $T_1 = s$. Then $\sum_{T\in U(s,q)} \prod_{1\leq i\leq q}(T_i+1)^{T_i+1} = (s+1)^{s+1} = 0.89^0(s+1)^{s+1}$.

Assume that the theorem holds for all sequences of length $q \geq 1$. We will show that it also holds for all sequences of length $q+1$.

$$\begin{aligned}\sum_{T\in U(s,(q+1))} \prod_{1\leq i\leq q+1}(T_i+1)^{T_i+1} &\leq \sum_{T_1=1}^{s-q}(T_1+1)^{T_1+1} \sum_{T'\in U((s-T_1),q)} \prod_{1\leq i\leq q}(T'_i+1)^{T'_i+1}\\ &\leq \sum_{a=1}^{s-1}(a+1)^{a+1}0.89^{q-1}(s-a+1)^{s-a+1}\\ &\leq 0.89^{q-1}\sum_{a=1}^{s-1}(a+1)^{a+1}(s-a+1)^{s-a+1}\\ &\leq 0.89^{q}(s+2)^{s+2}\end{aligned}$$

□

We can now bound the excess of the entire graph. We have that $\mathbf{ex}(G) \leq \sum_{i=1}^{m} \mathbf{ex}(C''_{v_i})$. In order for $\sum_{i=1}^{m} \mathbf{ex}(C''_{v_i}) \geq s$, there must be some sequence $j'_1, \ldots, j'_m$ such that $\sum_i j'_i \geq s$ and $\mathbf{ex}(C''_{v_i}) = j'_i$. Equivalently, if there was a sequence $j_1, \ldots, j_m$ such that $\sum_i j_i = s$ and $\mathbf{ex}(C''_{v_i}) \geq j_i$ then $\sum_{i=1}^{m} \mathbf{ex}(C''_{v_i}) \geq s$ would also be satisfied.

Therefore

$$\begin{aligned}
\mathbf{Pr}(\mathbf{ex}(G) \geq s) &\leq \mathbf{Pr}(\sum_{i=1}^{m} \mathbf{ex}(C''_{v_i}) \geq s) \\
&\leq \sum_{\substack{j_1,\ldots,j_m \\ \sum_i j_i = s}} \mathbf{Pr}(\wedge_i \mathbf{ex}(C''_{v_i}) \geq j_i) \\
&\leq \sum_{q=1}^{s} \sum_{\substack{j_1,\ldots,j_m \\ \sum_i j_i = s \\ |\{j_i : j_i \geq 1\}| = q}} \prod_{\{j_i : j_i \geq 1\}} \epsilon_2 \left(\frac{\epsilon_3 (j_i+1)}{m}\right)^{j_i+1} \\
&\leq \sum_{q=1}^{s} \binom{m}{q} \sum_{T \in U(s,q)} \prod_{\{x \in T\}} \epsilon_2 \left(\frac{\epsilon_3 (x+1)}{m}\right)^{x+1} \\
&\leq \sum_{q=1}^{s} \binom{m}{q} \epsilon_2^q \left(\frac{\epsilon_3}{m}\right)^{s+q} \sum_{T \in U(s,q)} \prod_{\{x \in T\}} (x+1)^{x+1} \\
&\leq \sum_{q=1}^{s} \binom{m}{q} \epsilon_2^q \left(\frac{\epsilon_3}{m}\right)^{s+q} 0.89^{q-1} (s+1)^{s+1} \\
&\leq \frac{1}{0.89} (s+1)^{s+1} \left(\frac{\epsilon_3}{m}\right)^{s} \sum_{q=1}^{s} \left(\frac{em}{q}\right)^{q} \epsilon_2^q \left(\frac{\epsilon_3}{m}\right)^{q} 0.89^q \\
&\leq \frac{1}{0.89} (s+1)^{s+1} \left(\frac{\epsilon_3}{m}\right)^{s} \sum_{q=1}^{s} \left(\frac{0.89 e \epsilon_2 \epsilon_3}{q}\right)^{q} \\
&\leq \frac{(s+1)e}{0.89} \left(\frac{\epsilon_3 s}{m}\right)^{s} \sum_{q=1}^{s} \left(\frac{0.89 e \epsilon_2 \epsilon_3}{q}\right)^{q}
\end{aligned}$$

The summation on the right results in a constant. Concretely, over the positive reals the function $\left(\frac{a}{x}\right)^x$ is maximized at $x = \frac{a}{e}$, for which it has value $e^{\frac{a}{e}}$. Therefore:

$$\begin{aligned}
\sum_{x=1}^{\infty} \left(\frac{a}{x}\right)^x &\leq \sum_{x=1}^{2a-1} \left(\frac{a}{x}\right)^x + \sum_{x=2a}^{\infty} \left(\frac{a}{x}\right)^x \\
&\leq \sum_{x=1}^{2a-1} (e^{\frac{a}{e}}) + \sum_{x=2a}^{\infty} \left(\frac{1}{2}\right)^x \\
&\leq e^{\frac{a}{e}}(2a-1) + 1 \\
&\leq 2ae^{\frac{a}{e}}
\end{aligned}$$

Therefore

$$\begin{aligned}\mathbf{Pr}(\mathbf{ex}(G) \geq s) &\leq \frac{(s+1)e}{2}\left(\frac{\epsilon_3 s}{m}\right)^s 4e\epsilon_2\epsilon_3 e^{2\epsilon_2\epsilon_3} \\ &\leq 2(s+1)e^2\epsilon_2\epsilon_3 e^{0.89\epsilon_2\epsilon_3}\left(\frac{\epsilon_3 s}{m}\right)^s\end{aligned}$$

Recall that $\epsilon_2 = \frac{4e}{1-\epsilon_0}$ and $\epsilon_3 = \frac{1-\epsilon_0}{e\epsilon_1^2}$ so $\epsilon_2\epsilon_3 = \frac{4}{\epsilon_1^2} = 4(2-\epsilon_0)^2\epsilon_0^{-4}$, so

$$\mathbf{Pr}(\mathbf{ex}(G(m,m,Po\left((1-\epsilon_0)m\right))) \geq s) \leq 2(s+1)e^2 4(2-\epsilon_0)^2\epsilon_0^{-4}e^{0.89*4(2-\epsilon_0)^2\epsilon_0^{-4}}\left(\frac{\epsilon_3 s}{m}\right)^s$$

Adding the probability that the Poissonization fails yields the final upper-bound:

Theorem 5. *The probability that two-table cuckoo hashing with n elements, tables of size $m = \frac{1+\epsilon_0}{1-\epsilon_0}n$ for $\epsilon_0 \in (0,1)$ and a stash of size s fails is:*

$$\boldsymbol{Pr}(\boldsymbol{ex}(G(m,m,n) \geq s+1) \leq \epsilon_4(s+2)\left(\frac{\epsilon_5(s+1)}{n}\right)^{s+1} + e^{-\epsilon_6 n}$$

where

$$\begin{aligned}\epsilon_4 &= 8(2-\epsilon_0)^2\epsilon_0^{-4}e^{2+0.89*4(2-\epsilon_0)^2\epsilon_0^{-4}} \\ \epsilon_5 &= \frac{(2-\epsilon_0)^2(1-\epsilon_0)^2}{e(1+\epsilon_0)\epsilon_0^4} \\ \epsilon_6 &= \epsilon_0 - \ln(1+\epsilon_0)\end{aligned}$$

When $n = \Omega(s)$, with an implicit constant of above ϵ_5, and $s = \omega(\log(N))$ this will be negligible in N. We therefore have a bound that shows that Cuckoo Hashing has negligible failure probability for any super-logarithmic stash sizes. As we have shown that logarithmic stash sizes cannot yield negligible failure probability, this analysis is tight in terms of the asymptotics of the stash sizes.

References

[ADW14] Martin Aumüller, Martin Dietzfelbinger, and Philipp Woelfel. Explicit and efficient hash families suffice for cuckoo hashing with a stash. *Algorithmica*, 70(3):428–456, 2014.

[DK12] Michael Drmota and Reinhard Kutzelnigg. A precise analysis of cuckoo hashing. *ACM Transactions on Algorithms (TALG)*, 8(2):1–36, 2012.

[GM11] Michael T Goodrich and Michael Mitzenmacher. Privacy-preserving access of outsourced data via oblivious RAM simulation. In *ICALP*, pages 576–587. Springer, 2011.

[GMOT12] Michael T Goodrich, Michael Mitzenmacher, Olga Ohrimenko, and Roberto Tamassia. Privacy-preserving group data access via stateless oblivious RAM simulation. In *SODA*, pages 157–167. SIAM, 2012.

[KLO12] Eyal Kushilevitz, Steve Lu, and Rafail Ostrovsky. On the (in) security of hash-based oblivious RAM and a new balancing scheme. In *SODA*, pages 143–156. SIAM, 2012.

[KMW09] Adam Kirsch, Michael Mitzenmacher, and Udi Wieder. More robust hashing: Cuckoo hashing with a stash. *SIAM Journal on Computing*, 39(4):1543–1561, 2009.

[LO13] Steve Lu and Rafail Ostrovsky. Distributed oblivious RAM for secure two-party computation. In *TCC*, pages 377–396. Springer, 2013.

[Pit98] Jim Pitman. Enumerations of trees and forests related to branching processes and random walks. *Microsurveys in discrete probability*, 41:163–180, 1998.

[PR01] Rasmus Pagh and Flemming Friche Rodler. Cuckoo hashing. In *ESA*, pages 121–133. Springer, 2001.