Parameterized Dynamic Tables

Tobias Nipkow[1]

Technische Universität München

Abstract

We analyze the amortized complexity of dynamic tables with arbitrary minimal and maximal load factors and arbitrary expansion and contraction factors.

Keywords: amortized complexity, dynamic tables, machine-checked proof

1. Introduction

One of the standard examples of amortized analysis are dynamic tables [2] that support insertion and deletion of elements. The size of the table is doubled when it overflows and it is halved when less than a quarter of the table is occupied.

For the analysis of dynamic tables we abstract the system state to a pair (n, l) where n is the number of elements in the table and l the size of the table. The quotient n/l is the *load factor*.

Generalized dynamic tables are parameterized by the minimal and maximal load factors (1/4 and 1 above) and the expansion and contraction factors (both 2 above). The aims of this paper are very modest: we do not present a new data structure for dynamic tables (e.g. [1]), we merely parameterize the standard dynamic tables and analyze in detail the amortized complexity of insertion and deletion as a function of these parameters.

The formal parameters are:

$f_1, f_2 > 0$ are the lower and upper bounds of the load factor

$e, c > 1$ are the expansion and contraction factors

Upon expansion, l is muliplied by e and upon contraction l is divided by c, and the invariant $f_1 \leq n/l \leq f_2$ must be maintained by expansion and contraction. Expanding a table with load factor f_2 leads to the load factor f_2/e, contracting a table with load factor f_1 leads to the load factor $f_1 c$. Thus we need $f_1 \leq f_2/e$ and $f_1 c \leq f_2$. In fact we require the strict versions

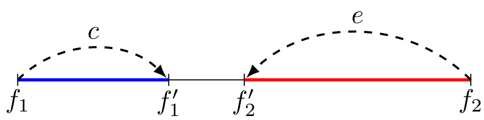
$$f_1 < f_2/e \quad (1)$$

$$f_1 c < f_2 \quad (2)$$

URL: `http://www.in.tum.de/~nipkow` (Tobias Nipkow)

[1]Research supported by DFG grant NI 491/16-1

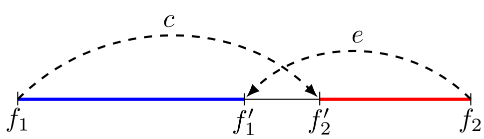
for reasons explained below. These are already the key constraints that guarantee constant amortized complexity for insertion and deletion. For an intuitive explanation look at the following typical behaviour of the load factor under contraction and expansion: it increases from f_1 to $f_1' := f_1 c$ and decreases from f_2 to $f_2' := f_2/e$.



Contraction and expansion need to copy $f_1 l$ and $f_2 l$ many elements. Spreading this cost over $(f_1' - f_1)l$ and $(f_2 - f_2')l$ leads to the following amortized costs for a single insertion and deletion step:

$$a_i := f_2/(f_2 - f_2') \qquad a_d := f_1/(f_1' - f_1) \tag{a_{id}}$$

However, there is second possible relationship among the parameters depicted in the following diagram:



To cater for both situations we need to set

$$f_1' := min\ (f_1 c)\ (f_2/e) \qquad f_2' := max\ (f_1 c)\ (f_2/e) \tag{f_{12}'}$$

rather than the naive setting of f_1' and f_2' before. From now on we assume the definitions (a_{id}) and (f_{12}'). Together with (1) and (2) it follows that

$$f_1'/c \leq f_1 \qquad f_2 \leq f_2' e \qquad f_1 \leq f_2'/c \qquad f_1' e \leq f_2$$

Furthermore note the following consequences of (1), (2), (a_{id}), (f_{12}'), $f_1 > 0$, and $e, c > 1$:

$$0 < f_1 < f_1' \leq f_2' < f_2 \qquad 1 < a_i \qquad 0 < a_d$$

Now we consider initialization. The standard dynamic tables start with $(0, 0)$, expand to $(1, 1)$ with the next insertion and only double the size afterwards. That precise schema no longer works for generalized dynamic tables. If $f_2 < 1$, the state $(1, 1)$ is not admissible. Thus we need an initial size l_0 such that $1 \leq f_2 l_0$ (and maybe other constraints on l_0). A second problem arises with non-integral e or c (or if $e \neq c$). Then for small table sizes contraction may let the load factor drop below f_1. For example, let $f_1 = 1/3$ and $c = 3/2$. Deleting an element from the state $(2, 6)$ requires contraction and leads to $(1, 4)$. Therefore we allow the load factor to drop below f_1 for small table sizes. We introduce the parameter l_0 and let the initial table be $(0, l_0)$. Contraction takes place only if $l \geq c l_0$, otherwise the load factor may drop below f_1.

The rest of the paper we analyze two models, where l is a real and one where l is a natural number.

Because we can model operations on tables as side effect free functions on pairs (n, l), we employ functional programming notation: functions with two parameters are written as $f\ x\ y$ rather than $f(x, y)$ and *if-then-else* returns either branch as a result.

2. Real length

In this version of the tables, n is a natural number but l is a non-negative real number. This abstracts a system where the actual table has size $\lfloor l \rfloor$, but we also carry around the real l that records the idealized length of the table, which can be non-integral if c or e are not integers or $c \neq e$. This allows us to focus on the core of the problem without the complications introduced by rounding.

First we define the behaviour of our generalized dynamic tables precisely. There are two operations *Ins* and *Del* whose behaviour is defined by the following next-state function *nxt*:

$$
\begin{aligned}
&\mathit{nxt}\ \mathit{Ins}\ (n,\ l) = (n + 1,\ \textsf{if}\ n + 1 \le f_2 l\ \textsf{then}\ l\ \textsf{else}\ el)\\
&\mathit{nxt}\ \mathit{Del}\ (n,\ l) =\\
&(n - 1,\ \textsf{if}\ f_1 l \le n - 1\ \textsf{then}\ l\ \textsf{else}\ \textsf{if}\ l_0 \le l/c\ \textsf{then}\ l/c\ \textsf{else}\ l)
\end{aligned}
$$

The behaviour of *Ins* is obvious; *Del* implements the decision to contract only if l does not drop below l_0. For natural numbers we assume $0 - 1 = 0$.

The execution time t is the number of elements that need to be copied:

$$
\begin{aligned}
&t\ \mathit{Ins}\ (n,\ l) = (\textsf{if}\ n + 1 \le f_2 l\ \textsf{then}\ 1\ \textsf{else}\ n + 1)\\
&t\ \mathit{Del}\ (n,\ l) = (\textsf{if}\ f_1 l \le n - 1\ \textsf{then}\ 1\ \textsf{else}\ \textsf{if}\ l_0 \le l/c\ \textsf{then}\ n\ \textsf{else}\ 1)
\end{aligned}
$$

2.1. Invariant

The invariant expresses that the table size does not drop below l_0, the load factor does not drop below f_1 as long as $cl_0 \le l$ and the load factor never exceeds f_2:

$$l_0 \le l \wedge (l_0 \le l/c \longrightarrow f_1 l \le n) \wedge n \le f_2 l \qquad \text{(Inv)}$$

However, l_0 needs to be large enough to guarantee invariance of (Inv):

$$1/(f_1(c - 1)) \le l_0 \qquad (3)$$
$$1/(f_2(e - 1)) \le l_0 \qquad (4)$$

Note that (4) implies $0 < l_0$ because $0 < f_2$ and $1 < e$.

Theorem 1. *Under the conditions* (1)–(4), (Inv) *is invariant.*

Proof. It is easy to see that the invariant holds in the initial state $(0,\ l_0)$.

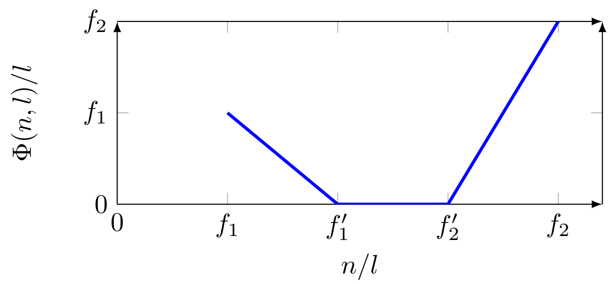
Now we prove that the invariant is preserved by insertion. If $n + 1 \le f_2 l$, this is trivial. Now assume $f_2 l < n + 1$, i.e., expansion. Of course $l_0 \le el$

because $l_0 \leq l$ and $1 < e$. From (1) it follows that $f_1(el) \leq f_2 l$ and thus $f_1(el) \leq n + 1$. From $n + 1 \leq f_2 l + 1$ (because $n \leq f_2 l$), (4) and $l_0 \leq l$ it follows that $n + 1 \leq f_2(el)$.

Finally we prove that the invariant is preserved by deletion. If $f_1 l \leq n - 1$, this is trivial. Now assume $n - 1 < f_1 l$. If $l/c < l_0$, the table does not contract and (Inv) is easily seen to be preserved. Now consider the contraction case where $l_0 \leq l/c$. Combining this with (3) yields $f_1(l/c) \leq f_1 l - 1$. Together with $f_1 l \leq n$ (because $l_0 \leq l/c$) we obtain $f_1(l/c) \leq n - 1$. From (2) we obtain $f_1 l \leq f_2(l/c)$ which implies $n - 1 \leq f_2(l/c)$ because $n - 1 < f_1 l$. $\square$

2.2. Amortized Complexity

The gist of the potential function is shown in the figure below:



The x axis shows the load factor, the y axis the potential normalized by l for load factors between f_1 and f_2. Going downward from f_1' and upward from f_2', the potential reaches $f_1 l$ and $f_2 l$, exactly enough to pay for the impending contraction and expansion. Between f_1' and f_2' the potential is 0.

This behaviour of Φ applies in the situation where $l/c \geq l_0$. If $l/c < l_0$, where we do not contract anymore, we set Φ to 0 between f_1 and f_1' as well. The precise and full definition if Φ is as follows:

$\Phi\ (n,\ l) =$
$(\textit{if}\ f_2' l \leq n\ \textit{then}\ a_i(n - f_2' l)$
$\ \textit{else if}\ n \leq f_1' l \wedge l_0 \leq l/c\ \textit{then}\ a_d(f_1' l - n)\ \textit{else}\ 0)$

For fixed l, Φ is of the form $\Psi\ n =$

$\textit{if}\ x_2 \leq n\ \textit{then}\ i(n - x_2)\ \textit{else if}\ n \leq x_1 \wedge b\ \textit{then}\ d(x_1 - n)\ \textit{else}\ 0$

The following lemma is proved by an easy exhaustive case analysis:

Lemma 1. *If* $i > 0$ *and* $d \geq 0$ *then* $\Psi\ (n{+}1) - \Psi\ n \leq i$.

The analogous lemma for deletion needs additional assumptions:

Lemma 2. *If* $i > 0$, $d \geq 0$, $n \neq 0$ *and* $x_1 \leq x_2$ *then* $\Psi\ (n{-}1) - \Psi\ n \leq d$.

Proof. The proof is a similar exhaustive case analysis. We consider only the case that requires $x_1 \leq x_2$. Assume $\neg\ x_2 \leq n - 1$, $n - 1 \leq x_1 \wedge \neg\ b$ and $x_2 \leq n$:

$$\begin{aligned}&\Psi\ (n{-}1) - \Psi\ n = d(x_1 - (n - 1)) - i(n - x_2)\\&\le d(x_1 - (n - 1)) = d + d(x_1 - n)\\&\le d \quad \text{because } x_1 \le x_2 \le n\end{aligned}$$ □

In order to prove the desired amortized complexity we need two more lower bounds for l_0 that guarantee that after an expansion or contraction there is at least one normal step that can pay for the next expansion or contraction, i.e., the intervals $[f_1,\ f_1']$ and $[f_2',\ f_2]$ must be large enough:

$$1/((f_1' - f_1)c) \le l_0 \tag{5}$$
$$1/(f_2 - f_2') \le l_0 \tag{6}$$

Theorem 2. *Under the conditions (1)–(6), the amortized complexity of insertion and deletion is $\le a_i + 1$ and $a_d + 1$.*

Proof. It is easy to see that potential of the initial state $(0,\ l_0)$ is 0. Moreover, the potential is clearly never negative.

Now we assume that (Inv) holds and let $s = (n,\ l)$. First we show

$$A := t\ \mathit{Ins}\ s + \Phi\ (\mathit{nxt}\ \mathit{Ins}\ s) - \Phi\ s \le a_i + 1 \tag{$*$}$$

In the non-expansion case, i.e., if $n + 1 \le f_2 l$, $t\ \mathit{Ins}\ s = 1$ and $\mathit{nxt}\ \mathit{Ins}\ s = (n + 1,\ l)$. Therefore Lemma 1 applies and thus $A = 1 + a_i$. Now assume $f_2 l < n + 1$. Then $t\ \mathit{Ins}\ s = n + 1$. Moreover, $\Phi\ s = a_i(n - f_2' l)$ because $f_2' l \le n$ because $f_2 l < n + 1$ and $1 \le (f_2 - f_2')l$ because $l_0 \le l$ and (6). By case analysis we prove $A \le n - a_i(f_2 - f_2')l + a_i + 1$, which implies $A \le n - f_2 l + a_i + 1$ (by def. of a_i) and hence ($*$) because $n \le f_2 l$ (Inv). If $f_2'(el) \le n + 1$ then $\Phi\ (\mathit{nxt}\ \mathit{Ins}\ s) = a_i(n + 1 - f_2'(el))$. Therefore

$$\begin{aligned}&A = n + 1 + a_i(n + 1 - f_2'(el)) - a_i(n - f_2' l)\\&= n - a_i(f_2' e - f_2')l + a_i + 1\\&\le n - a_i(f_2 - f_2')l + a_i + 1 \quad \text{because } f_2 \le f_2' e\end{aligned}$$

Now assume $n + 1 < f_2'(el)$. From $f_1' e \le f_2$ it follows that $f_1'(el) \le f_2 l$. Together with $f_2 l < n + 1$ this yields $f_1' el < n + 1$ and thus $\Phi\ (\mathit{nxt}\ \mathit{Ins}\ s) = 0$. Thus:

$$\begin{aligned}&A = n + 1 - a_i(n - f_2' l)\\&= n - a_i(n + 1 - f_2' l) + a_i + 1\\&\le n - a_i(f_2 - f_2')l + a_i + 1 \quad \text{because } f_2 l < n + 1\end{aligned}$$

Thus we have proved ($*$). Now we show

$$A := t\ \mathit{Del}\ s + \Phi\ (\mathit{nxt}\ \mathit{Del}\ s) - \Phi\ s \le a_d + 1 \tag{$**$}$$

If $n = 0$ then (Inv) implies $l/c < l_0$ and thus $t\ \mathit{Del}\ s = 1$, $\Phi\ s = 0$, $\mathit{nxt}\ \mathit{Del}\ s = s$ and thus $A = 1$. We treat $n = 0$ separately because if $0 < n$ the expression $n - 1$ does not require a case analysis to take care of $0 - 1 = 0$. Now assume $0 < n$. In the non-contraction cases, i.e., if $l/c < l_0$ or $f_1 l \le n - 1$, $t\ \mathit{Del}\ s = 1$ and $\mathit{nxt}\ \mathit{Del}\ s = (n - 1,\ l)$. Therefore Lemma 2 applies (because $f_1' \le f_2'$) and thus $A \leq 1 + a_d$. Now we consider the contraction case: $n - 1 < f_1 l$ and thus $t\ Del\ s = n$. From (5) and $l_0 \leq l/c$ it follows that $1 \leq (f_1' - f_1)l$ and thus $n < f_1' l$ because $n - 1 < f_1 l$. Hence $\Phi\ s = a_d(f_1' l - n)$ using $f_1' \leq f_2'$. By case analysis we prove $A \leq n - a_d(f_1' - f_1)l + a_d$, which implies $A \leq n - f_1 l + a_d$ (by def. of a_d) and hence $(**)$ because $n - 1 < f_1 l$. In case $n - 1 < f_1'(l/c) \wedge l_0 \leq l/c/c$ then $\Phi\ (nxt\ f\ s) = a_d(f_1' l/c - (n - 1))$, because $n - 1 < f_2'(l/c)$ (because $n - 1 < f_1 l$ and $f_1 \leq f_2'/c$). Thus $A = n + a_d(f_1'(l/c) - (n - 1)) - a_d(f_1' l - n) = n + a_d(f_1'/c - f_1')l + a_d \leq n + a_d(f_1 - f_1')l + a_d$ (because $f_1'/c \leq f_1$). In case $\neg\ (n - 1 < f_1'(l/c) \wedge l_0 \leq l/c/c)$ then $\Phi\ (nxt\ f\ s) = 0$ again because $n - 1 < f_2'(l/c)$. Thus $A = n + a_d(n - f_1' l) = n + a_d(n - 1 - f_1' l) + a_d \leq n - a_d(f_1' - f_1)l + a_d$ (because $n - 1 < f_1 l$). $\square$

2.3. Optimal Parameters

The fact that Φ is 0 between f_1' and f_2' is clearly suboptimal. The optimal situation is $f_1' = f_2'$, i.e., $f_1 c = f_2/e$. Thus we can eliminate one of our four parameters and we choose f_1:

$$f_1 = f_2/(ec) \tag{7}$$

In this case one constraint for l_0 suffices:

$$ec/(f_2(min\ e\ c - 1)) \leq l_0 \tag{8}$$

From (8) we obtain immediately that

$$ec/(f_2(e - 1)) \leq l_0 \tag{9}$$
$$ec/(f_2(c - 1)) \leq l_0 \tag{10}$$

Because $e,c > 1$, this implies (3) and (4). Thus we have the following corollary to Theorem 1.

Corollary 1. *Under conditions* (1), (2), (7) *and* (8), (Inv) *is invariant.*

Optimality requires

$$f_1' = f_2' = f_2/e. \tag{11}$$

Together with (9) this implies (6), and together with (10) this implies (5) (because $e,c > 1$). Thus we have the following corollary to Theorem 2:

Corollary 2. *Under conditions* (1), (2), (7) *and* (8), *the amortized complexity of insertion and deletion is* $\leq a_i + 1$ *and* $a_d + 1$ *where* $a_i = e/(e - 1)$ *and* $a_d = 1/(c - 1)$.

The values for a_i and a_d follow directly from the definitions of a_i and a_d together with (7) and (11). The following table shows a few concrete optimal parameter settings (with minimal values for l_0).

f_1	f_2	e	c	l_0
1/4	1	2	2	4
1/5	4/5	2	2	5
1/10	9/10	3	3	5
1/3	1	3/2	2	6

3. Integer length

In this version of the tables, n, l and l_0 are natural numbers. This is the standard model of dynamic tables. However, we do not assume that e and c are the same integers. Hence expansion and contraction may require conversions to integers. We modify *nxt* and (Inv) by replacing el by $\lceil el \rceil$ and l/c by $\lfloor l/c \rfloor$. In fact, (Inv) remains unchanged because $l_0 \le \lfloor l/c \rfloor \longleftrightarrow l_0 \le l/c$ because l_0 is an integer.

3.1. Invariant

The invariant is no longer preserved under the original constraints, i.e., Theorem 1 no longer holds. For expansion ($f_2 l < n + 1$) and contraction ($n - 1 < f_1 l$) we could prove $f_1(el) \le n + 1$ and $n - 1 \le f_2(l/c)$ but $f_1\lceil el \rceil \le n + 1$ and $n - 1 \le f_2\lfloor l/c \rfloor$ do not necessarily hold. However, two more lower bounds for l_0 solve the problem:

$$f_1/(f_2 - f_1 e) \le l_0 \tag{12}$$
$$f_2/(f_2 - f_1 c) \le l_0 \tag{13}$$

Theorem 3. *Under the conditions* (1)–(4)*,* (12) *and* (13)*,* (Inv) *is invariant.*

Proof. It is easy to see that the invariant holds in the initial state.

Now we prove that the invariant is preserved by insertion. If $n + 1 \le f_2 l$, this is trivial. Now assume $f_2 l < n + 1$, i.e., expansion. From (12), (1) and $l_0 \le l$ we obtain $f_1 \le (f_2 - f_1 e)l$ and thus $f_1\lceil el \rceil \le f_1(el + 1) \le f_2 l < n + 1$. From $n + 1 \le f_2 l + 1$ (because $n \le f_2 l$), (4) and $l_0 \le l$ it follows that $n + 1 \le f_2(el) \le f_2\lceil el \rceil$.

Finally we prove that the invariant is preserved by deletion. If $f_1 l \le n - 1$, this is trivial. Now assume $n - 1 < f_1 l$. If $l/c < l_0$, the table does not contract and (Inv) is easily seen to be preserved. Now consider the contraction case where $l_0 \le l/c$. Combining this with (3) yields $f_1(l/c) \le f_1 l - 1$. Together with $f_1 l \le n$ (because $l_0 \le l/c$) we obtain $f_1\lfloor l/c \rfloor \le f_1(l/c) \le n - 1$. From (13), (2) and $l_0 \le l/c$ we obtain $f_1 l \le f_2(l/c - 1)$ and thus $n - 1 < f_1 l \le f_2(l/c - 1) \le f_2\lfloor l/c \rfloor$. $\square$

We also need to increase the amortized complexity of insertion and deletion. In the proof of Theorem 2 we could show that after an expansion, the load factor must be $\ge f_1'$ and after a contraction it must be $\le f_2'$. This is no longer the case due to floor and ceiling and cannot be fixed by increasing l_0 either. The problem is that the load factor may drop just a little below f_1' or rise a little above f_2'. To compensate for this we widen the interval $[f_1', f_2']$.

First we analyze a system parameterized by two constants f_1'' and f_2'' subject to certain constraints. Then we define f_1'' and f_2'' and prove that for large enough l_0 the constraints are satisfied. Essentially, f_1'' and f_2'' take the place of f_1' and f_2' but should be a bit smaller/larger:

$$f_1 < f_1'' < f_1' \qquad f_2' < f_2'' < f_2 \tag{14}$$
$$1 \le (f_1'' - f_1)cl_0 \tag{15}$$
$$1 \le (f_2 - f_2'')l_0 \tag{16}$$

Constraints (15) and (16) replace (5) and (6). Of course a_i and a_d are redefined analogously:

$$a_i = f_2/(f_2 - f_2'') \qquad a_d = f_1/(f_1'' - f_1)$$

In addition we need

$$l_0 \le l \Longrightarrow f_1''(l+1) \le f_1' l \tag{17}$$
$$l_0 \le l \Longrightarrow f_2' l \le f_2''(l-1) \tag{18}$$

Theorem 4. *Under the conditions (1)–(4), (12)–(18), the amortized complexity of insertion and deletion is* $\le a_i + 1$ *and* $a_d + 1$*.*

Proof. We redefine Φ by replacing f_1' and f_2' by f_1'' and f_2''. The non-expansion and non-contraction cases of the proof are the same as in the proof of Theorem 2. Although the definition of Φ has changed, Lemmas 1 and 2 still apply because a_i, $a_d > 0$, $f_1' \le f_2'$ and (14).

Assume (Inv) and $s = (n, l)$. First we prove the expansion case ($f_2 l < n + 1$, $t\ Ins\ s = n + 1$) of

$$A := t\ Ins\ s + \Phi\ (nxt\ Ins\ s) - \Phi\ s \le a_i + 1 \tag{$*$}$$

From (16) and $l_0 \le l$ we obtain $1 \le (f_2 - f_2'')l$ and thus $f_2'' l \le n$ because $f_2 l < n + 1$. Thus $\Phi\ s = a_i(n - f_2'' l)$. By case analysis we prove $A \le n - a_i(f_2 - f_2'')l + a_i + 1$, which implies $A \le n - f_2 l + a_i + 1$ (by def. of a_i) and hence ($*$) because $n \le f_2 l$ (Inv). If $f_2''\lceil el\rceil \le n + 1$ then $\Phi\ (nxt\ Ins\ s) = a_i(n + 1 - f_2'(el))$. Therefore

$$\begin{aligned}
A &= n + 1 + a_i(n + 1 - f_2''\lceil el\rceil) - a_i(n - f_2' l) \\
&= n - a_i(f_2''\lceil el\rceil - f_2'' l) + a_i + 1 \\
&\le n - a_i(f_2 - f_2'')l + a_i + 1 \quad \text{because } f_2 \le f_2' e \text{ and (14)}
\end{aligned}$$

Now assume $n + 1 < f_2''\lceil el\rceil$. We have $f_1''\lceil el\rceil \le f_1''(el + 1) \le f_1'(el) \le f_2 l \le n + 1$ by $l_0 \le el$, (17) and $f_1' e \le f_2$. Together with $n + 1 < f_2''\lceil el\rceil$ this implies $\Phi\ (nxt\ Ins\ s) = 0$. Therefore

$$\begin{aligned}
A &= n + 1 - a_i(n - f_2'' l) \\
&= n - a_i(n + 1 - f_2'' l) + a_i + 1 \\
&\le n - a_i(f_2 - f_2'')l + a_i + 1 \quad \text{because } f_2 l < n + 1
\end{aligned}$$

Thus we have proved ($*$). Now we show for the contraction case ($n - 1 < f_1 l$ and $l_0 \le \lfloor l/c\rfloor$ and thus $t\ Del\ s = n$) that

$$A := t\ Del\ s + \Phi\ (nxt\ Del\ s) - \Phi\ s \le a_d + 1 \tag{$**$}$$

From (15), $l_0 \le l/c$ and $f_1 < f_1''$ it follows that $1 \le (f_1'' - f_1)l$ and thus that $n < f_1'' l$ because $n - 1 < f_1 l$. Moreover $f_1'' \le f_2''$ because of (14) and $f_2' \le$

f_1'. Hence $\Phi\ s = a_d(f_1''l - n)$. We also have $n - 1 < f_2''\lfloor l/c \rfloor$: $n - 1 < f_1 l \le f_2'(l/c) \le f_2''(l/c - 1) \le f_2''\lfloor l/c \rfloor$ because $f_1 \le f_2'/c$, (18) and $l_0 \le l/c$. By case analysis we prove $A \le n - a_d(f_1'' - f_1)l + a_d$, which implies $A \le n - f_1 l + a_d$ (by def. of a_d) and hence $(**)$ because $n - 1 < f_1 l$. In case $n - 1 < f_1''\lfloor l/c \rfloor \wedge l_0 \le \lfloor \lfloor l/c \rfloor /c \rfloor$ then $\Phi\ (\mathit{nxt}\ f\ s) = a_d(f_1''\lfloor l/c \rfloor - (n - 1))$ because $n - 1 < f_2''\lfloor l/c \rfloor$, $n - 1 < f_1 l$ and $l_0 \le \lfloor l/c \rfloor$. Therefore

$$\begin{aligned}
&A = n + a_d(f_1''\lfloor l/c \rfloor - (n - 1)) - a_d(f_1''l - n)\\
&\le n + a_d(f_1''(l/c) - (n - 1)) - a_d(f_1''l - n)\\
&= n - a_d(f_1'' - f_1''/c)l + a_d\\
&\le n - a_d(f_1'' - f_1)l + a_d \quad \text{by (14) and } f_1'/c \le f_1
\end{aligned}$$

In case $\neg\ (n - 1 < f_1''\lfloor l/c \rfloor \wedge l_0 \le \lfloor \lfloor l/c \rfloor /c \rfloor)$ then $\Phi\ (\mathit{nxt}\ f\ s) = 0$ because $n - 1 < f_2''\lfloor l/c \rfloor$, $n - 1 < f_1 l$ and $l_0 \le \lfloor l/c \rfloor$. Therefore

$$\begin{aligned}
&A = n - a_d(f_1''l - n)\\
&= n - a_d(f_1''l - (n - 1)) + a_d\\
&\le n - a_d(f_1'' - f_1)l + a_d \quad \text{because } n - 1 \le f_1 l
\end{aligned}$$

□

Now we show that the constraints for f_1'' and f_2'' are satisfied for certain definitions of f_1'' and f_2'' given certain additional lower bounds for l_0:

Lemma 3. *Let* $f_1'' = f_1'l_0/(l_0 + 1)$ *and* $f_2'' = f_2'l_0/(l_0 - 1)$. *Then*

$$(f_2 + 1)/(f_2 - f_2') \le l_0 \tag{19}$$
$$(f_1'c + 1)/((f_1' - f_1)c) \le l_0 \tag{20}$$
$$f_1/(f_1' - f_1) < l_0 \tag{21}$$
$$f_2/(f_2 - f_2') < l_0 \tag{22}$$

imply (14) – (18)

Proof. First note that $1 < l_0$ because $f_2' < f_2$ and (22). By definition of f_1'' it follows that $f_1'' < f_1'$ and that $f_1 < f_1'' \longleftrightarrow f_1/(f_1' - f_1) < l_0$ and hence that $f_1 < f_1''$ because of (21). Similarly we obtain $f_2' < f_2''$ and $f_2'' < f_2$ from $1 < l_0$ and (22). Thus we have proved (14). From (20), $f_1 < f_1'$ and the def. of f_1'' we obtain $1 \le (f_1' - f_1)cl_0 - f_1'c = (f_1'(l_0 - 1)/l_0 - f_1)cl_0 \le (f_1'l_0/(l_0 + 1) - f_1)cl_0 = (f_1'' - f_1)cl_0$ and thus (15). From the def. of f_2'', $1 < l_0$, (19) and $f_2' < f_2$ it follows that $1 \le (f_2 - f_2'')(l_0 - 1)$ and thus (16). For the proof of (17) assume $l_0 \le l$. Together with $0 < l_0$ and the def. of f_1'' this implies $f_1''(l + 1) = f_1'(l_0/(l_0 + 1))(l + 1) \le f_1'(l/(l + 1))(l + 1) = f_1'l$ and thus (17). For the proof of (18) assume $l_0 \le l$. Together with $1 < l_0$ and the def. of f_2'' this implies $f_2'l = f_2'l + f_2'((l_0 - 1)/(l_0 - 1) - 1) \le f_2'l + f_2'((l - 1)/(l_0 - 1) - 1) = f_2''(l - 1)$ and thus (18). □

4. Conclusion

The preceding analysis is not very deep mathematically but full of cases and details. In order to guard against errors, all proofs [4] have been checked with the help of the Isabelle theorem prover by building on a framework for amortized analysis proofs [3].

5. References

[1] A. Brodnik, S. Carlsson, E. D. Demaine, J. I. Munro, and R. Sedgewick. Resizable arrays in optimal time and space. In F. K. H. A. Dehne, A. Gupta, J. Sack, and R. Tamassia, editors, *Algorithms and Data Structures, WADS'99*, volume 1663 of *LNCS*, pages 37–48. Springer, 1999.

[2] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, 3rd edition, 2009.

[3] T. Nipkow. Amortized complexity verified. In C. Urban and X. Zhang, editors, *Interactive Theorem Proving (ITP 2015)*, volume 9236 of *LNCS*. Springer, 2015. http://in.tum.de/~nipkow/pubs/itp15.html.

[4] T. Nipkow. Parameterized dynamic tables. *Archive of Formal Proofs*, 2015. http://afp.sf.net/entries/Dynamic_Tables.shtml, Formal proof development.