Correctness of KACTL's modmul

Simon Lindholm

2020-05-02

1 Introduction

Within computational number theory, and for hashing, there is sometimes a need to compute modular multiplications $a \cdot b \pmod{c}$ for relatively large c, in particular larger than 2^{32}. KACTL contains the following algorithm for computing this for $0 \le a, b \le c < 7.268 \cdot 10^{18}$: [1]

```
typedef int64_t ll;
typedef uint64_t ull;
ull mod_mul(ull a, ull b, ull c) {
    ll ret = a * b - c * ull(1.L / c * a * b);
    return ret + c * (ret < 0) - c * (ret >= (ll)c);
}
```

It assumes an x86 or x86_64 processor with long doubles compiled to use 80-bit x87 float registers (as on e.g. GCC), and runs about 2 times faster than the naive expression `(__int128_t)a * b % c`. This paper shows why it works.

On a historical note, an earlier version of KACTL included the same algorithm but with the slightly modified expression `ull((long double)a * b / c)`. This can be shown precise all the way up to $c < 2^{63}$; however, it is slightly longer and slightly slower in the common case of performing several modular multiplications with the same c, due to the division.

2 The basic idea

$a \cdot b \pmod{c} = ab - \lfloor ab/c \rfloor c$. We can compute the value ab/c approximately using floating point numbers – in this case 80-bit long doubles, as seen by `1.0L` in the code. Letting $R \approx ab/c$, $S = ab - \lfloor R \rfloor c$ will be a number that's congruent to ab modulo c, while being relatively close to the desired range $[0, c)$. To get it into the target range we simply add c if the result of the computation is negative, or subtract c if it is greater than or equal to c. It is fine for the computations ab and $\lfloor R \rfloor c$ to overflow – arithmetic will be performed mod 2^{64} and the residue when converted into $[-2^{63}, 2^{63})$ will be the value we reduce in $[0, c)$.

For the algorithm to work we will need to prove two things:

[1] this number equals $r \cdot 2^{64}$, where $r = (\sqrt{177} - 7)/16 \approx 0.394$ is the positive solution to the equation $8x^2 + 7x = 4$.

1. S is in $[-c, 2c)$
2. S is in $[-2^{63}, 2^{63})$

The second one of these will be where the bound comes from.

3 S is in $[-c, 2c)$

When performing a basic arithmetic operation (addition, subtraction, multiplication, division) $\oplus$ on two long doubles a, b, the resulting long double will be $r(a \oplus b)$, where $r(x)$ denotes rounding x to the nearest long double, with ties broken in favor of the one with a trailing zero in the bit representation.

80-bit x87 long doubles are represented with a sign bit, a 15-bit exponent, and a 64-bit mantissa, of which the topmost bit is always 1. It can represent the integers $0, 1, \ldots, 2^{64}$ perfectly, but the next representable integer after that is $2^{64} + 2$. Thus, for x in $[2^{64}, 2^{65})$, the difference between x and $r(x)$ is at most 1, and this rescales similarly to other powers of two in the exponent range that we will be working with. In particular, we have the inequality $|x - r(x)| \leq x \cdot 2^{-64}$. By abuse of notation, we will write this as $r(x) = x \cdot (1 \pm 2^{-64})$, with $\pm a$ representing any number in the range $[-a, a]$.

Now, let us consider the expression $S = ab - \lfloor r(r(r(1/c)a)b) \rfloor c$, which we want to prove is in the range $[-c, 2c)$. Flooring subtracts less than one, so this would be implied by

$$ab - r(r(r(1/c)a)b)c \in [-c, c]$$

which we rewrite as

$$|ab/c - r(r(r(1/c)a)b)| \leq 1.$$

If $c \leq 2^{62}$, we have $r(r(r(1/c)a)b) = ab/c \cdot (1 \pm 2^{-64})^3$, yielding

$$\begin{aligned} |ab/c - r(r(r(1/c)a)b)| &\leq (3 \cdot 2^{-64} + 3 \cdot 2^{-128} + 2^{-192}) \cdot ab/c \\ &< 4 \cdot 2^{-64} \cdot 2^{62} = 1. \end{aligned}$$

Otherwise:

- $2^{-63} < 1/c < 2^{-62}$, so $r(1/c) = 1/c \pm 2^{-127}$,
- $r(1/c)a < 1.001 \cdot a/c < 2$, so $r(r(1/c)a) = r(1/c)a \pm 2^{-64}$,
- $r(r(1/c)a)b < 1.001 \cdot ab/c < 2^{63}$, so $r(r(r(1/c)a)b) = r(r(1/c)a)b \pm 2^{-2}$,

and hence

$$\begin{aligned} r(r(r(1/c)a)b) &= ((1/c \pm 2^{-127})a \pm 2^{-64})b \pm 2^{-2} \\ &= ab/c \pm 2^{-127}ab \pm 2^{-64}b \pm 2^{-2}, \end{aligned}$$

yielding a difference from the exact value of at most

$$2^{-127}c^2 + 2^{-64}c + 2^{-2} \leq 0.313 + 0.395 + 0.25 = 0.958 < 1$$

given $c \leq 0.395 \cdot 2^{64}$.

4 S is in $[-2^{63}, 2^{63})$

Since $c < 2^{63}$, we get the bound $-2^{63} \leq S$ from the previous one. If $c \leq 2^{62}$, we also get the latter one. However, the case where $c > 2^{62}$ requires some care. Let us proceed by contradiction and assume $S \geq 2^{63}$. If we manage to use this to deduce $c \geq X$ we will know contrapositively that the bound holds for $c < X$. Expanding S, the assumption we have is that

$$ab - \lfloor r(r(r(1/c)a)b) \rfloor c \geq 2^{63}$$

We can weaken this assumption by making the floor part smaller. $r(x)$ is a monotonically increasing function and all numbers are non-negative, so in particular we can make subexpressions of it smaller.

If $r(1/c)a \geq 1$, then we can successively weaken the inequality:

$$\begin{aligned}2^{63} &\leq ab - \lfloor r(r(1) \cdot b) \rfloor c \\ &= ab - bc \\ &\leq bc - bc \\ &= 0\end{aligned}$$

and get a contradiction. Otherwise, $r(1/c)a < 1$, so $r(r(1/c)a) \geq r(1/c)a - 2^{-65}$.

As in the previous section, $2^{62} < c < 2^{63}$ implies $r(1/c) \geq 1/c - 2^{-127}$.

Since $0 \leq r(r(1/c)a)b < 1.0001c < 2^{64}$, all integers near it are exactly representable, and applying r can't move it past an integer. Thus, $\lfloor r(r(r(1/c)a)b) \rfloor > r(r(1/c)a)b - 1$. Combining these inequalities results in

$$\begin{aligned}2^{63} &\leq ab - \lfloor r(r(r(1/c)a)b) \rfloor c \\ &\leq ab - (((1/c - 2^{-127})a - 2^{-65})b - 1)c \\ &\leq 2^{-127}abc + 2^{-65}bc + c.\end{aligned}$$

Substituting $a = 2^{64}x, b = 2^{64}y, c = 2^{64}z$, we can rewrite this as $1/2 \leq 2xyz + 1/2 \cdot yz + z$ with $0 \leq x, y \leq z$. By using $0 \leq x, y \leq z$ and solving the equation $1/2 = 2z^3 + 1/2 \cdot z^2 + z$, we see that this implies $z \geq 0.351$. This is not quite what we were shooting for, though – our aim is 0.394.

To go above this, we need to improve upon the bound for $\lfloor r(r(r(1/c)a)b) \rfloor$. Let k be such that $r(r(1/c)a)b$ is in the range $[2^k, 2^{k+1})$. Then if it its distance to the next larger integer is less than 2^{k-64}, it will round upwards before being floored. Hence, flooring can only reduce the value by at most $1 - 2^{k-64}$, so we get the bound $\lfloor r(r(r(1/c)a)b) \rfloor \geq r(r(1/c)a)b - 1 + 2^{k-64}$.

Depending on a, b, c we may end up with different values of k. The maximal k is achieved by $a = b = c$, where $k = 62$. For this k, we get a similar bound to before, except with $1 - 2^{-2}$ instead of 1: $1/2 \leq 2xyz + 1/2yz + 3/4 \cdot z$. Using

$x, z \leq z$ and solving for equality yields $z \geq 0.3962$, which implies the bound we want.

For $k = 61$, $r(r(1/c)a)b < 2^{62}$, which implies that ab/c is similarly bounded. Loosely, we get $ab/c < 1.0001 \cdot r(r(1/c)a)b < 1.0001 \cdot 2^{62}$, and so

$$\begin{aligned} 2^{63} &\leq 2^{-127}abc + 2^{-65}bc + 7/8 \cdot c \\ &= 2^{-127}ab/c \cdot c^2 + 2^{-65}bc + 7/8 \cdot c \\ &< 1.0001 \cdot 2^{62} \cdot 2^{-127} \cdot c^2 + 2^{-65}bc + 7/8 \cdot c \\ &\leq 1.0001 \cdot 2^{-64}c^2 + 7/8 \cdot c \end{aligned}$$

This solves to around $z = 0.394$, the bound we want. We will get back to this case for a more careful analysis, without the loose 1.0001 factor.

For $k = 60$, the same argument gives $2^{63} \leq 0.7501 \cdot 2^{-64}c^2 + 15/16 \cdot c$ which solves to $z = 0.403$, more than enough.

For $k \leq 59$, we get $2^{63} \leq 0.6251 \cdot 2^{-64}c^2 + c$, which solves to $z = 0.399$, also enough.

It remains to analyze the $k = 61$ case in more detail, and show that the bound does hold for all z up to the root of $1/2 = z^2 + 7/8z$. If $a/c > 0.999$, b would need to be small for ab/c to be below 2^{62}, causing the $2^{-65}bc$ term to shrink enough that we get a (much) better bound than 0.394 even with the 1.0001 slop factor. Otherwise, $r(1/c)a < 1$, so $r(1/c)a \leq r(r(1/c)a) + 2^{-65}$. Instead of the loose inequality, we write

$$\begin{aligned} ab/c &= (1/c)ab \\ &\leq (r(1/c) + 2^{-127})ab \\ &= r(1/c)ab + 2^{-127}ab \\ &\leq (r(r(1/c)a) + 2^{-65})b + 2^{-127}ab \\ &= r(r(1/c)a)b + 2^{-65}b + 2^{-127}ab \\ &< 2^{62} + 2^{-65}b + 2^{-127}ab/c \cdot c \\ &\leq 2^{62} + 2^{-65}c + 2^{-127} \cdot 2^{62} \cdot 1.0001 \cdot c \\ &\leq 2^{62} + 0.395 \cdot 2^{64} \cdot (2^{-65} + 2^{-65}) \\ &= 2^{62} + 0.395. \end{aligned}$$

Hence,

$$\begin{aligned} 2^{63} &\leq 2^{-127}ab/c \cdot c^2 + 2^{-65}bc + 7/8 \cdot c \\ &< (2^{62} + 0.395) \cdot 2^{-127} \cdot c^2 + 2^{-65}bc + 7/8 \cdot c \\ &\leq (2^{-64} + 0.395 \cdot 2^{-127})c^2 + 7/8 \cdot c. \end{aligned}$$

Solving this gives a value of c that floors to the same integer bound (to be exact, 7268172458553106874) as the same equation without the epsilon term, which thus be removed. This finishes the proof.