make10

Stephen Huan

https://cgdct.moe

Theory club 2023-09-08

Overview

make10

Figure: *Suugaku Josi* (*Math Girls*) by Masae Yasuda

Some examples

First example

2, 4, 6, 8

$$2 + 6 + 8/4 = 10$$

Second example

6, 2, 4, 5

$$(6 + 4 - 5) \times 2 = 10$$

Third example

3, 4, 7, 8

Initial observations

Easy to prove (just give construction)

Hard to disprove (prove impossible?)

Computer-assisted proofs

Try to prove sequences of numbers can $\mathrm{make10}$

Fibonacci sequence

Definition

Fibonacci sequence F_n defined recursively:

$$F_0 := 0$$
$$F_1 := 1$$
$$F_n = F_{n-1} + F_{n-2}.$$

The first few terms are $0, 1, 1, 2, 3, 5, 8, 13, \ldots$

A claim

Theorem

$\mathrm{make10}([F_i]_{i=1}^{n})$ *if and only if* $n > 3$.

A claim

Theorem

$\mathrm{make10}([F_i]_{i=1}^n)$ *if and only if* $n > 3$.

Proof.

"Base" cases:

$n = 0$: $\emptyset$.

$n = 1$: 1.

$n = 2$: $1, 1$.

$n = 3$: $1, 1, 2$.

All impossible.

A claim

Theorem

$\mathrm{make10}([F_i]_{i=1}^n)$ *if and only if* $n > 3$.

Proof.

Base cases:

$n = 4$: $1, 1, 2, 3.$

$$(1 + 1 + 3) \times 2.$$

$n = 5$: $1, 1, 2, 3, 5.$

$$(3 - 2) \times 5 \times (1 + 1).$$

$n = 6$: $1, 1, 2, 3, 5, 8.$

$$1 + 1 + 2 + 3 - 5 + 8.$$

Found contiguous $n \equiv 1, 2, 0 \pmod 3$.

A claim

Theorem

$\mathrm{make10}([F_i]_{i=1}^{n})$ *if and only if* $n > 3$.

Proof.

By induction.
Let $n = 3k + r$ for $k > 1$, $0 \leq r < 3$.
By definition $F_{n+2} = F_{n+1} + F_n$, so $F_{n+2} - F_{n+1} - F_n = 0$.
Take

$$[F_{n+2} - F_{n+1} - F_n] + [F_{n-1} - F_{n-2} - F_{n-3}] + \cdots + \mathrm{make10}([F_i]_{i=1}^{k\in\{4,5,6\}})$$

depending on r (pick $k \equiv r \pmod 3$). □

Commentary

How reliant on structure was the result?

How far can we push this?

$F_n \sim \phi^n$ where $\phi := (1 + \sqrt{5})/2$ is the golden ratio

First prove (surprisingly) useful lemma...

Lemma

For any integer $x \neq 0$, $\mathrm{make10}([x]_{i=1}^{n})$ *if* $n > 8$.

Constant

Lemma

For any integer $x \neq 0$, $\text{make10}([x]_{i=1}^{n})$ *if* $n > 8$.

Proof.

Follow previous sketch.

$n = 9$:

$$(x + x) \times (x + x + x + x + x)/x/x = (2x)(5x)/x^2 = 10$$

$n = 10$:

$$(x/x + x/x) \times (x + x + x + x + x)/x = (2)(5x)/x = 10$$

Continuous in $\pmod 2$.

Lemma

For any integer $x \neq 0$, $\mathrm{make10}([x]_{i=1}^{n})$ *if* $n > 8$.

Proof.

By induction again.
Take $(x - x) + (x - x) + \cdots + \mathrm{make10}([x]_{i=1}^{k \in \{9,10\}})$. □

Lemma

For any integer $x \neq 0$, $\mathrm{make10}([x]_{i=1}^{n})$ *if* $n > 8$.

Proof.

By induction again.
Take $(x - x) + (x - x) + \cdots + \mathrm{make10}([x]_{i=1}^{k\in\{9,10\}})$. □

Conjecture.

There exists a constant X such that for all $x > X$, $\mathrm{make10}([x]_{i=1}^{n})$ if and only if $n > 8$.

Corollary

For any integer $x \neq 0$, $\mathrm{make10}([x^i]_{i=1}^{n})$ *if* $n > 16$.

Corollary

For any integer $x \neq 0$, $\mathrm{make10}([x^i]_{i=1}^n)$ *if* $n > 16$.

Proof.

Take adjacent quotients of pairs of $[x, x^2, ..., x^n]$, from right to left.
Let $n = 2k$ or $2k + 1$ depending on parity of n.
We get $[x, ..., x]$, a list of k or $(k + 1)$ x's depending on parity of n.
But we know $\mathrm{make10}$ of a list of k constants is possible if $k > 8$.
So we need $k > 8$ or $n > 16$. □

Corollary

$\mathrm{make10}([i]_{i=1}^{n})$ *if* $n > 16$.

Corollary

$\mathrm{make10}([i]_{i=1}^{n})$ *if* $n > 16$.

Proof.

Take adjacent differences of pairs of $[1, 2, \ldots, n]$, from right to left. The proof follows identically, with $x = 1$. □

Corollary

$\mathrm{make10}([i]_{i=1}^{n})$ *if* $n > 16$.

Proof.

Take adjacent differences of pairs of $[1, 2, \ldots, n]$, from right to left. The proof follows identically, with $x = 1$. □

Note.

Since $x = 1$, it's possible to sharpen the lemma to

$n = 7$: $(1 + 1) \times (1 + 1 + 1 + 1 + 1)$.

$n = 8$: $(1 + 1) \times (1 + 1 + 1 + 1 + 1) \times 1$.

This means we can improve the bound to $k > 7$, or $n > 14$. This is still not even close to tight.

Theorem

$\mathrm{make10}([i]_{i=1}^{n})$ *if and only if* $n > 3$.

Optimal linear

Theorem

$\text{make10}([i]_{i=1}^n)$ *if and only if* $n > 3$.

Proof.

Base cases:

$n = 1$. Impossible.

$n = 2$. Impossible.

$n = 3$. Impossible.

$n = 4$. $1 + 2 + 3 + 4$.

$n = 5$. $(1 + 2 + 3 - 4) \times 5$.

$n = 6$. $(6 - 5) \times (1 + 2 + 3 + 4)$.

Theorem

$\mathrm{make10}([i]_{i=1}^{n})$ *if and only if* $n > 3$.

Proof.

For $n > 6$ consider:

$$[(n-3)-(n-1)]\times\left[\frac{1+2+\cdots+(n-6)}{n-5}+(n-4)-(n-2)\right]+n$$

Clearly every number from $1,\ldots,n$ is included. Simplifying,

$$\begin{aligned}&=(-2)[((n-6)(n-5)/2)/(n-5)-2]+n\\&=-(n-6)+4+n\\&=10\end{aligned}$$

which was to be proven. □

Towards arbitrary polynomials

We would like to strengthen this result

Most natural generalization to monomials i^k

Would like to prove something of the flavor

Theorem

For any natural k there exists a N such that for all $n > N$, $\mathrm{make10}([i^k]_{i=1}^n)$. (Note that N is allowed to depend on k.)

Definition

The *discrete derivative* of a sequence of n numbers $f : \{1, \ldots, n\} \to \mathbb{R}$ is the new sequence of $n-1$ numbers defined by $\mathrm{d}f(i) := f(i+1) - f(i)$ for $1 \leq i < n$.

Theorem

d *is a linear operator on* f*.*

Lemma

Recall an operator d *is said to be linear if*

$$\mathrm{d}(f+g) = \mathrm{d}f + \mathrm{d}g$$
$$\mathrm{d}(cf) = c(\mathrm{d}f)$$

for all $f, g \in \mathbb{R}^n$ *and* $c \in \mathbb{R}$*.*

Derivative of a monomial

Lemma

If $f(i) = ci^k$ *for some natural* k *and real* c*, then* $\mathrm{d}f$ *is a* $(k-1)$*-degree polynomial in* i *with leading coefficient* kc*.*

Derivative of a monomial

Lemma

If $f(i) = ci^k$ for some natural k and real c, then $\mathrm{d}f$ is a $(k-1)$-degree polynomial in i with leading coefficient kc.

Proof.

Use the binomial theorem

$$(i+1)^k = \sum_{j=0}^{k} \binom{k}{j} i^j 1^{k-j} = \sum_{j=0}^{k} \binom{k}{j} i^j.$$

$$\mathrm{d}f(i) = c[(i+1)^k - i^k] = c \sum_{j=0}^{k-1} \binom{k}{j} i^j.$$

where we used that when $j = k$, $\binom{k}{k} i^k = i^k$.
The leading term is $c\binom{k}{k-1} i^{k-1} = cki^{k-1}$ as claimed. □

Derivative of a polynomial

Lemma

If $f(i)$ is a k-degree polynomial in i with leading coefficient c, then $\mathrm{d}f$ is a $(k-1)$-degree polynomial in i with leading coefficient kc.

Derivative of a polynomial

Lemma

If $f(i)$ is a k-degree polynomial in i with leading coefficient c, then $\mathrm{d}f$ is a $(k-1)$-degree polynomial in i with leading coefficient kc.

Proof.

Apply the previous lemma term-by-term with the linearity of d.
For all terms i^j with $j < k$, derivative has power $< k - 1$.
Thus the only term which affects the leading coefficient is i^k, by the previous lemma, derivative has leading term cki^{k-1}, as claimed. □

Repeated derivatives

Lemma

If $f(i) = ci^k$ *for natural* k *and real* c*, then* $\mathrm{d}^k f = k!$.

Repeated derivatives

Lemma

If $f(i) = ci^k$ for natural k and real c, then $\mathrm{d}^k f = k!$.

Proof.

Repeatedly apply the previous lemma until left with a constant. □

Proving the claim

Theorem

For any natural k there exists a N such that for all $n > N$, $\text{make10}([i^k]_{i=1}^n)$.

Proving the claim

Theorem

For any natural k there exists a N such that for all $n > N$, $\mathrm{make10}([i^k]_{i=1}^n)$.

Definition

Define the operator $\mathrm{d}_2 f$ on $f \in \mathbb{R}^n$ as

$$\mathrm{d}_2 f(i) := \begin{cases} \mathrm{d}f[\mathsf{begin} : 2 : \mathsf{end}] & n \equiv 0 \pmod 2 \\ [f[\mathsf{begin}], \mathrm{d}f[\mathsf{begin} + 1 : 2 : \mathsf{end}]] & n \equiv 1 \pmod 2 \end{cases}$$

in `Julia` slice notation.

Proving the claim

Theorem

For any natural k there exists a N such that for all $n > N$, $\text{make10}([i^k]_{i=1}^n)$.

Proof.

Since $\mathrm{d}f$ is a $(k-1)$-degree polynomial, $\mathrm{d}_2 f$ is a also a $(k-1)$-degree polynomial, possibly with a "Dirac" at the beginning.

Derivatives of Diracs

The derivative of a Dirac at $i = 1$ stays a Dirac (with flipped sign). This is not true for a Dirac placed in the middle!

We can essentially ignore the effect of the Dirac (linearity).

Proving the claim

Theorem

For any natural k there exists a N such that for all $n > N$, $\mathrm{make10}([i^k]_{i=1}^n)$*.*

Proof.

Note that $|\mathrm{d}f(i)| \leq |f(i+1)| + |f(i)|$.
Since each value is represented exactly once in d_2, $\|\mathrm{d}_2 f\|_1 \leq \|f\|_1$.
By induction, $\left\|\mathrm{d}_2^k f\right\|_1 \leq \|f\|_1$, so $|\delta| \leq \left|1^k + \cdots + n^k\right| < \mathcal{O}(n^{k+1})$ where $|\delta|$ denotes the size of the Dirac.
If we have $[x, \ldots, x, y]$, we can trivially do this in $\mathcal{O}(|y|)$ x's by

$$y - [x/x + x/x + \cdots + x/x].$$

Since $|\delta|$ grows with n, this leads to infinite regress (we need more terms to control the Dirac, which leads to the Dirac growing).

Proving the claim

Theorem

For any natural k there exists a N such that for all $n > N$, $\mathrm{make10}([i^k]_{i=1}^n)$.

Proof.

We can get out of this by either finding a tighter bound on $|\delta|$ or a more efficient algorithm for controlling the Dirac. We will do both.
Write y in binary. Form 2 by $(x+x)/x$, to get 2^i we need i copies.
For i digits, need $\mathcal{O}(i^2)$ x's or $\mathcal{O}(\log^2(|y|))$. Is this enough?
We can take $n > \mathcal{O}(2^{(1+o(1))k})$. But we can do better.
We can bound $\left|\mathrm{d}_2^k f(i)\right| \leq 2^{\binom{k}{2}} k!$. Since this is independent of n, we can take $n > \mathcal{O}(2^{k^2})$ which is massive, but workable. With the more efficient algorithm, we can take $n > \mathcal{O}(k^2 2^k)$. □

Expressivity of polynomials

Theorem

(Expressivity of polynomials) For any sequence f on n terms, there exists a $(n-1)$-degree polynomial p such that $f(i) = p(i)$.

Proof.

Each i defines a linear constraint in the n coefficients of p.
There are n points and n coefficients defining a $n \times n$ linear system.
This system is solvable if and only if all of the points (the i's) are distinct, which they necessarily are. □

Commentary

Certificate requires $\widetilde{\mathcal{O}}(2^k)$ terms for a k-degree polynomial

We'd need $\widetilde{\mathcal{O}}(2^n)$ terms for an arbitrary sequence of length n

Trade-off between "complexity" of sequence (as measured by polynomial degree) and "training data" (number of terms)

From this perspective, $\mathrm{make10}$ measures the difficulty of the learning problem in modeling/understanding the sequence.

Conjectures

conjecture

$\text{make10}([F_n, F_{n+1}, F_{n+2}, F_{n+3}])$ is impossible for $n > 5$.

conjecture

$\text{make10}([n, n, n, n])$ is impossible for $n > 20$.
(checked by computer up to $n = 10^5$)

conjecture

$\text{make10}([n, n+1, n+2, n+3])$ is impossible for $n > 22$.
(checked by computer up to $n = 10^5$)

conjecture

For any natural k there exists a N such that for all $n > N$,
$\text{make10}([n^k, (n+1)^k, (n+2)^k, (n+3)^k])$ is impossible.

Conclusion

Children's game to improve mental arithmetic

Study how to $\mathrm{make10}$ with longer sequences

How to prove $\mathrm{make10}$ is impossible?

Computer-assisted proofs

Just now...!!?

Julia code I

```julia
import Base.show


const RATIONAL = false

divide(x, y) = RATIONAL ? x // y : x / y

const OPERATORS = [
    ((x, y) -> x + y, '+', false),
    ((x, y) -> x - y, '-', false),
    ((x, y) -> y - x, '-', true),
    ((x, y) -> x * y, '*', false),
    ((x, y) -> divide(x, y), '/', false),
    ((x, y) -> divide(y, x), '/', true),
]

const LEAF = '.'


struct Tree{T}
    value::T
    left::Union{Tree{T},Nothing}
    right::Union{Tree{T},Nothing}
    name::Char
end


Tree(x) = Tree(divide(x, one(x)), nothing, nothing, LEAF)

isleaf(x) = x.name == LEAF

```

Julia code II

```julia
isunit(x) = !in(x.name, ['+', '-'])

const PAREN_TABLE = Dict(
    '+' => Pair((_) -> true, (_) -> true),
    '-' => Pair((_) -> true, isunit),
    '*' => Pair(isunit, isunit),
    '/' => Pair(isunit, isleaf),
)

function show(io::IO, tree::Tree)
    value = if isleaf(tree)
        Int(tree.value)
    else
        left = repr(tree.left; context=io)
        right = repr(tree.right; context=io)
        left_paren, right_paren = PAREN_TABLE[tree.name]
        left_repr = !left_paren(tree.left) ? "($left)" : left
        right_repr = !right_paren(tree.right) ? "($right)" : right
        "$left_repr $(tree.name) $right_repr"
    end
    return print(io, value)
end


function key(nums)
    return Tuple(sort([tree.value for tree in nums]))
end

make10(nums) = make10!(collect(map(Tree, nums)))
```

Julia code III

```julia
function make10!(nums; cache=Set())
    n = length(nums)
    n == 0 && return nothing
    n == 1 && nums[begin].value == 10 && return nums[begin]
    for i in 1:n
        for j in i:(n - 1)
            for (op, name, flip) in OPERATORS
                x = popat!(nums, i)
                y = popat!(nums, j)
                value = op(x.value, y.value)
                pushfirst!(nums, Tree(value, flip ? y : x, flip ? x : y, name))
                k = key(nums)
                if !in(k, cache)
                    push!(cache, k)
                    rtn = make10!(nums; cache)
                    # no need to maintain the invariant
                    !isnothing(rtn) && return rtn
                end
                # restore order
                popfirst!(nums)
                insert!(nums, j, y)
                insert!(nums, i, x)
            end
        end
    end
end

@assert !isnothing(make10([2, 4, 6, 8])) "failed example 1"
@assert !isnothing(make10([6, 2, 4, 5])) "failed example 2"
println(make10([3, 4, 7, 8]))
```