

BEAF 맛보기

이상학

개요

googology는 그 이름에서 알 수 있듯이 거대한 수들에 대해 연구하는 유희 수학의 한 분야입니다. BEAF는 고기가 아니라 Bowers' Exploding Array Function의 약자로, 큰 수를 나타내는 표기법의 일종이죠. 간단하면서도 상당히 강력한 체계로 유명합니다. 이러한 작업은 수학적으로 의미가 깊지 않지만 (그리고 뇌절을 좀 많이 치기는 하지만) 많은 흥미로운 이야기들을 포함하고 있습니다. 본 세미나는 그러한 심연에 가볍게 발을 담귀볼 기회를 갖는 자리입니다.

0 도입

googology는 큰 수를 연구하는 분야. 모든 수단을 동원하여 가능한 한 큰 수를 우아하게 표현하기 위한 작업. 정말, 정말 많은 이야기가 있지만, 모두 얘기하기엔 시간이 없으니 빨리 본제로 들어가자.

BEAF는 Bowers' Exploding Array Function의 약어이다. 배열을 통해 큰 수를 나타내는 배열 기반 표기법. 잘 짜인 재귀 구조를 바탕으로 수의 규모를 폭발시킨다. 후반부의 정의는 오류가 있는 것으로 밝혀졌지만, 잘 정의된 것만 해도 충분히 강력. fgh 기준 ϵ_0 단계... 대충 페아노 공리계만큼 강력하다.

googology에서 항상 쓰이는 기법인, '연산을 자기 자신만큼 반복'하여 크기를 향상시킨다. 세부 단계는 다음과 같다.

- 윗화살표 표기법
- 선형 배열 표기법
- 다차원 배열 표기법
- 테트레이션 배열 표기법

1 윗화살표 - 선형 배열

커누스 윗화살표 표기법

가장 기본적인 것부터, 곱셈은 덧셈의 반복이다.

$$a \times b = \underbrace{a + a + \cdots + a}_b$$

그리고 지수는 곱셈의 반복:

$$a^b = \underbrace{a \times a \times \cdots \times a}_b$$

증가율을 생각하면 덧셈 $\ll$ 곱셈 $\ll$ 지수.

다음 단계는 테트레이션, 지수의 반복이다.

$${}^b a = \underbrace{a^{a^{\cdots^a}}}_b$$

지수는 결합법칙이 성립하지 않으므로, 오른쪽부터 계산한다. 이는 지수보다 훨씬 큰 수를 표기할 수 있다(ex. ${}^3 3$). 이에 대한 흥미로운 이야기도 많지만... 넘어가자.

다음 단계로 펜테이션 등을 정의할 수 있다. 3 펜테이션 3은 ${}^3 3 =$ 엄청 큰 수... 하지만 슬슬 표기법이 부족해진다.

그래서 사용하는 윗화살표 표기법. 지수를 $a^b = a \uparrow b$ 로 표기한다. 그러면

$$a \uparrow\uparrow b = {}^a b = \underbrace{a \uparrow a \uparrow \cdots \uparrow a}_b$$

물론 연산은 오른쪽부터.

그러면 펜테이션은

$$a \uparrow\uparrow\uparrow b = \underbrace{a \uparrow\uparrow a \uparrow\uparrow \cdots \uparrow\uparrow a}_b$$

로 쉽게 정의 가능. 일반적으로, n 번째 윗화살표 연산은 다음과 같이 정의.

$$a \uparrow b = a^b$$

$$a \overbrace{\uparrow \cdots \uparrow}^{n+1} b = \underbrace{a \overbrace{\uparrow \cdots \uparrow}^n a \overbrace{\uparrow \cdots \uparrow}^n \cdots \overbrace{\uparrow \cdots \uparrow}^n a}_b$$

예를 들어

$$5 \uparrow \uparrow \uparrow 3 = 5 \uparrow \uparrow (5 \uparrow \uparrow 5) = 5 \uparrow \uparrow (5 \uparrow 5 \uparrow 5 \uparrow 5 \uparrow 5)$$

등으로, 벌써 상상치 않게 크다.

여담이지만 $a \uparrow b = a \times \cdots \times a$ 임을 고려하면 $\times$ 는 $n = 0$ 번째 윗화살표 연산으로, 같은 방식으로 $+$ 는 -1 번째 윗화살표 연산으로 간주할 수도 있다.

근데 n 이 커지면 다시 표기법이 번잡해지기 시작한다. 그러니 $\overbrace{\uparrow \cdots \uparrow}^n = \uparrow^n$ 으로 축약하자. 또한 재귀 구조를 분석하여 정의를 간소화:

$$a \uparrow^1 b = a^b, \quad a \uparrow^{n+1} 1 = a, \quad a \uparrow^{n+1} (b+1) = a \uparrow^n (a \uparrow^{n+1} b)$$

확장

이제 그 유명한 그레이엄 수를 정의할 수 있다. 일단 $g_1 = 3 \uparrow^4 3 = 3 \uparrow \uparrow \uparrow \uparrow 3$ 인데, 이거부터 가늠이 잘 안 된다. 그리고 $g_2 = 3 \uparrow^{g_1} 3$, $g_3 = 3 \uparrow^{g_2} 3$, ...으로 진행된다. 그렇게 g_{64} 까지 가면 그레이엄 수가 된다. 풀어서 쓰면

$$\begin{array}{c} 3 \uparrow \cdots \uparrow 3 \\ \underbrace{} \\ 3 \uparrow \cdots \uparrow 3 \\ \underbrace{} \\ \vdots \\ \underbrace{} \\ 3 \uparrow \uparrow \uparrow 3 \end{array}$$

이며 이게 64층이 있는 거.

이렇게 큰 수는 조합론에서 튀어나왔다. n 차원 큐브 그래프의 간선을 두 색으로 채색할 때, 항상 한 평면 위에 있는 동색 K_4 를 부분 그래프로 갖도록 하는 최소 n 의 상계. 이는 수학적으로 의미 있는 가장 큰 수로 기네스북에 등재된 것으로 유명하다. 물론 조합론의 몇몇 수는 (가늠이 안 될 정도로) 이보다 훨씬 크며, googology에는 더욱 깊은 심연이 존재한다. 그리고 이 상계는 현재 $2 \uparrow \uparrow \uparrow 5$ 수준으로 떨어져 상대적으로 덜(?) 크다.

그레이엄 수가 큰 수들 중에서는 작은 편이긴 하지만 이는 커누스 윗화살표를 초월하는 방법을 제시한다. 그레이엄 수는 확실히 윗화살표 표기법으로 간략히 나타낼 수 있는 웬만한 수보다 크다. 따라서 이런 식의 화살표 개수 재귀를 형식화하는 표기법을 도입한다면, 다음 단계의 큰 수로 나아갈 수 있다.

정의를 어떻게 하면 좋을까? 그레이엄 수의 매개변수는 세 개다: 기본이 되는 수 3, 초기 화살표 개수 4, 층위의 수 64. 그러나 지금까지 계속해서 두 개의 변수(기본 a 와 반복 횟수 b)만을 사용했으므로, 이에 맞춰 초기 화살표 개수도 기본으로 설정하자.

이를 위해 기호 $\uparrow$ 를 사용하면, 다음과 같은 정의를 얻는다:

$$a \uparrow b = a \underbrace{\uparrow \cdots \uparrow}_b a$$

여기서 이게 b 층 있는 것이다. 이전과 같이 재귀 구조를 분석하여 단순화하면,

$$a \uparrow 1 = a, \quad a \uparrow (b + 1) = a \uparrow^{a \uparrow b} a$$

이제 그레이엄 수는 $3 \uparrow 65$ 와 $3 \uparrow 66$ 사이에 있을 뿐이다.

커누스 윗화살표 표기법이 단순히 이변수 함수 $a \uparrow b$ 에 재귀를 거쳐 나온 큰 수 표기법의 계층임을 상기하자. 그러면 $a \uparrow b$ 보다 훨씬 거대한 $a \uparrow b$ 를 기반으로 같은 작업을 반복하면 그 이상의 계층을 얻을 수 있다. 즉,

$$a \uparrow\uparrow b = \underbrace{a \uparrow a \uparrow \cdots \uparrow}_b a$$

$$a \uparrow\uparrow\uparrow b = \underbrace{a \uparrow\uparrow a \uparrow\uparrow \cdots \uparrow\uparrow}_b a$$

$$\vdots$$

그리고 이것을 반복하여 $\uparrow \cdots \uparrow$ 를 $\uparrow^n$ 으로 축약하고 $\uparrow$ 에서 $\uparrow$ 를 뽑아내듯 $\uparrow$ 의 개수에 대한 재귀를 사용하면...? 뭐, $\uparrow$ 가 나오겠지. 그리고 이 과정을 또 반복할 수 있고...

자, 이제 어느 정도 한계에 다다랐다. 표기법을 정리해보자. 표기의 편의를 위해 a 와 b 사이에 작대기 d 개의 화살표가 c 개 있는 것을 $a \widehat{|d|}^c b$ 로 표기하겠다.

$$a \widehat{|d|}^c 1 = a, \quad a \widehat{|1|}^1 b = a^b$$

$$a \widehat{|d+1|}^1 (b+1) = a \widehat{|d|}^{a \widehat{|d+1|}^1 b} a$$

$$a \widehat{|d|}^{c+1} (b+1) = a \widehat{|d|}^c (a \widehat{|d|}^{c+1} b)$$

기본적으로 이 표기법은 d 에 대한 귀납, c 에 대한 귀납, b 에 대한 귀납으로 정의된다.

선형 배열 표기법

좀 더 표준적이고 보기 편하게, 배열 표기법으로 바꿔보자. $a \widehat{d}^c b$ 를 길이 4의 선형 배열 $\{a, b, c, d\}$ 로 표기하는 것이다.

$$\begin{aligned}\{a, 1, c, d\} &= a, & \{a, b, 1, 1\} &= a^b \\ \{a, b, 1, d\} &= \{a, a, \{a, b-1, 1, d\}, d-1\} \\ \{a, b, c, d\} &= \{a, \{a, b-1, c, d\}, c-1, d\}\end{aligned}$$

사실 이 단계부터 수의 규모 감이 정말 안 온다.

하지만 그렇더라도 계속 나아갈 수는 있다. 자세히 보면, 길이를 5로 아주 자연스레 확장할 수 있지 않은가?

$$\begin{aligned}\{a, 1, c, d, e\} &= a, & \{a, b, 1, 1, 1\} &= a^b \\ \{a, b, 1, 1, e\} &= \{a, a, a, \{a, b-1, 1, 1, e\}, e-1\} \\ \{a, b, 1, d, e\} &= \{a, a, \{a, b-1, 1, d, e\}, d-1, e\} \\ \{a, b, c, d, e\} &= \{a, \{a, b-1, c, d, e\}, c-1, d, e\}\end{aligned}$$

그러면 $\{a, 2, 1, 1, 2\} = \{a, a, a, a, 1\}$ 로, 지금까지 봐온 모든 수보다 큰 것을 쉽게 만들 수 있다.

이 규칙은 상당히 단순하기에 동일한 패턴에 따라 일반적인 유한 배열에 대한 계산 규칙을 명시할 수 있다. 용어 정리:

- base: 배열 맨 앞의 원소
- prime: 배열의 두 번째 원소
- pilot: 배열이 세 번째 이후 원소 중 처음으로 1이 아닌 것

그러면 배열 표기법의 계산 규칙은 다음과 같다. b 가 base이고 p 를 prime이라 하자.

- $p = 1$ 이면 b 이다.
- pilot이 없으면 b^p 이다.
- 아니면, pilot 바로 앞의 성분을 ‘배열 자신의 복제에서 prime을 1 줄인 것’으로 대체하고, 나머지 앞쪽을 모두 base로 바꾸고, pilot을 1 줄인다.

이 세 규칙만으로 웬만한 표기법은 가볍게 뛰어넘는 큰 수 표기법을 만들 수 있다.

대충 설명하면 base는 기본이 되는 수, prime은 메인 반복자, 나머지 뒤에는 메인 반복자를 폭발시키기 위한 보조 반복자들, 느낌이다. 더 이상 이게 얼마나 큰 자연수를 나타내는지는 설명할 수 없다. 대신에 서수를 사용하면 일종의 재귀 구조 단계 같은 걸 말할 수 있다... 참고로 지금은 ω^ω 수준

2 다차원 - 테트레이션

이제부터 끔찍하게 계속 이어지는 뇌절이 시작된다. 이제 수가 얼마나 큰지 파악하는 건 관두고 ‘아 그렇구나’ 하고 흘려들길 바람. 더 큰 수를 표기하기 위한 방법은 성장률을 폭증시키는 부분에 인자를 넣는 것.

다차원 배열 표기법

지금은 성장률이 ‘배열의 길이’에 따라 증가하므로, $b \& a = \overbrace{\{a, a, \dots, a\}}^b$ 로 표기하자. 이는 굉장히 성장률이 빠른 이변수 함수인데, 우리는 이변수 함수 a^b 를 엄청난 성장률을 가진 배열 표기법으로 확장하는 방법을 알고 있다. 그러니 배열 표기법 버전 2, $\{a, b\}_2 = b \& a$ 로 하고 나머지 규칙을 똑같이 두면?

하하, 버전 2가 있으면 $\{a, b\}_3 = \overbrace{\{a, a, \dots, a\}}^b_{\{a, a, \dots, a\}}$ 로 시작하는 버전 3도 있고, 계속 그렇게 할 수 있겠다. 아예 $\{a, a, \dots, a\}_{\{a, a, \dots, a\}}$ 로 시작하는 건? 이렇게 밑첨자를 b 번이나 내린 건? 이 지금까지의 모든 걸 뛰어넘는 단계를 $\{a, b\}_{1,2}$ 로 표기하면 굉장히 합리적이다.

왜인가? $\{a, 3\}_{1,2}$ 를 고려하자. 그러면 a 가 base, 3이 prime, 2가 (대충) pilot이다. 위의 선형 배열 규칙을 아날로그해서 그대로 적용하면

$$\{a, 3\}_{1,2} = \{a, 3\}_{\{a, 2\}_{1,2,1}} = \{a, 3\}_{\{a, 2\}_{\{a, 1\}_{1,2,1}}} = \{a, 3\}_{\{a, 2\}_a}$$

규칙을 그대로 써도 원하는 효과를 거둘 수 있음을 알았으니, 그냥 밑첨자를 배열에 포함시키면 어떨까? $\{a, b\}_{1,2}$ 는 $\{a, b(1)1, 2\}$ 로 표기한다. (1)은 1차원 심표, 정도로 해석하면 된다.

1차원 심표 뒤의 숫자들은 그 앞의 1차원 배열 구조의 길이를 폭발시킨다. 즉, 다음 규칙을 추가한다.

- 만약 pilot 앞에 1차원 심표가 있다면 그 앞쪽을 prime 개의 base로 채우고, 나머지 pilot의 앞을 base로 채운 후, pilot을 1 줄인다.

그러면 아래첨자 표기의 성질을 그대로 쓸 수 있다. 예를 들어,

$$\{a, 3(1)1, 1, 2\} = \{a, a, a(1)a, \{a, 2(1)1, 1, 2\}\} = \{a, a, a(1)a, \{a, a(1)a, a\}\}$$

좋다. 다음은 $\{a, \dots, a(1)a, \dots, a\}$ 일 것이며, 명백히 이는 $\{a, b(1)1(1)2\}$ 로 표기하는 게 맞겠다. 이렇게 1차원 쉼표를 여러 개 쓸 수 있다면 그 다음은 $\{a, \dots, a(1)a, \dots, a(1) \dots (1)a, \dots, a\}$ 이며, a 가 b^2 개 있는 것이다. 이것을 2차원 쉼표를 사용하여 $\{a, b(2)2\}$ 로 표기하는 것은 당연한 일이다. 이제 위 규칙은

- 만약 pilot 앞에 n 차원 쉼표가 있다면 그 앞쪽을 prime^n 개의 (적절히 쉼표를 넣은) base로 채우고...

음, 정의를 명백히 쓸 수도 있겠지만 솔직히 지금 그래봤자 지루하기만 하니 넘어가자. 이 다차원 배열을 밀어붙인 결과는 $\omega^{\omega^{\omega}}$ 단계의 귀납 구조를 갖는다.

테트레이션 배열 표기법

아직 안 끝났다... base를 b , prime을 p 라 하면 b 를 p^n 개 적절히 늘어놓아 엄청 큰 수를 표기할 수 있었다. n 이 커질수록 성장률이 커진다. 그러면 b 를 p^p 개 늘어놓는다면? 즉, $\{b, p(p)2\}$ 를 얻고 싶다.

배열 표기법의 규칙을 고려하여, 이를 $\{b, p(0, 1)2\}$ 로 표기한다. 이젠 엄밀히 말하는 걸 포기하겠다. 다음은 $\{b, p(1, 1)2\}$ 일 텐데 예를 들어

$$\{4, 3(1, 1)2\} = \{..(0, 1)..(0, 1)..\}$$

이며 여기서 ..은 3^3 개의 4이다. 즉 $(1, 1)2$ 는 b 를 p^{p+1} 개 늘어놓는다.

$(0, 2)2$ 는 $(p, 1)2$ 가 될 것이며 이는 p^{2p} 개의 b 이다. 일반적으로 $\{b, p(n, m)2\}$ 는 p^{n+mp} 개의 b 가 있다. 그러면 $\{b, p(0, 0, 1)2\} = \{b, p(p, p)2\}$ 로 쉼표 안에 배열을 집어넣을 수 있다.

다음 단계가 눈에 보이는가? 배열을 차원 배열로 확장할 수 있었다. 즉, $\{b, p(0(1)1)2\} = \{b, p(p, \dots, p)2\}$ 이게 $\omega \uparrow\uparrow 4$ 단계이며, 대충 $p^{p^p} = p \uparrow\uparrow 3$ 개의 b 가 있다.

물론 뇌절을 더할 수 있고, 쉼표 안의 배열의 차원을 늘리고, 계속 나아가서, 쉼표 안의 배열 안의 쉼표 안에 배열을 넣고, 이를 반복할 수 있다. 이렇게 하여 제대로 정의된 BEAF의 성장률 한계에 도달한다. 이는 $\varepsilon_0 = \omega \uparrow\uparrow \omega$ 단계로 표기한다.

휴.

3 여담

더 나아가

원래 BEAF의 정의는 여기서 더 나아간다. 지금 보면 ‘ b 가 몇 개인지’로 성장률을 늘려왔다. $\{b, p(0(1)1)2\}$ 는 $p \uparrow\uparrow 3$ 개, $\{b, p(0(0(1)1)1)2\}$ 는 $p \uparrow\uparrow 4$ 개, 그런 식이다. 그러면 이걸 뛰어넘으려면 $p \uparrow\uparrow p$ 개의 b 를 깔면 되지 않는가? 아니 $p \uparrow\uparrow\uparrow p$ 개는 어떨까? BEAF가 윗화살표 표기법의 일반화임을 고려하면, $\{p, p, p, p\}$ 개의 b 는 어떨까? 이런 식으로 미친듯이 나아갈 수 있다.

하지만 문제가 있다. 예컨대 p^2 개의 b 는, 2차원으로 깔린 b 들을 의미하는가, 단순히 p^2 개의 선형 b 배열을 의미하는가? 이를 제대로 명시하지 않았기에 이 이후의 BEAF는 모두 잘못 정의되었다. 이를 개선한 BAN 등의 표기법이 있다. BEAF를 완전히 제대로 정의하는 것은 googology의 큰 과제 중 하나이다.

엄밀히 정의해보기

테트레이션부터 정의를 아주 대충하기 시작했는데, 엄밀한 정의를 시도해보자. 우선 배열을 무엇으로 생각할 것이냐면, 무한히 존재하는 차원 구조의 공간? 에 유한 개의 자연수가 위치한 것이다. 무한한 수의 칸이 있고, (1)로 구분되는 그 다음 행에 또 무한한 수의 칸이 있고, 그 다음 행도, 다음 행도, ... 그렇게 무한히 많은 행이 있다. 그 다음 (2)로 구분되는 다음 평면에 똑같이 무한한 칸을 갖는 무한한 행이 있고, 또 다음 평면이 있고, 그렇게 존재하는 3차원 블록 너머엔 다음 블록이 있고... 뭐 그런 거다. 그런 구조 다음엔 (0, 1)로 구분되는 다음 구조가 또 있고, 하하.

이제 이러한 구조 공간의 앞부분에 이름을 붙인다. 자연수 m 에 대해 첫 m 개 칸은 m 구조라 하며, 첫 행 전체를 X 구조라 한다. 첫 행과 두 번째 행의 첫 m 개 칸은 $X + m$ 구조, 첫 행과 두 번째 행 전체는 $2X$ 구조, ..., 첫 평면은 X^2 구조, ...인 것이다. 그리고 그렇게 무한한 차원을 통틀어서, (0, 1) 직전까지의 전체를 X^X 구조라 한다. 이러한 ‘구조’를 귀납적으로 정의할 수도 있다. 아니면 그냥 추상적으로 X 에 대한 된 지수-다항식이다.

다음은 구조에 숫자를 채워넣는 법이다. α 가 구조일 때, $\alpha \&b[p]$ 는 대충 b 를 $\alpha(p)$ 개만큼 잘 배열한 것이다. 예를 들어

$$X^2 + X + 3\&b[2] = b, b(1)b, b(2)b, b(1)b, b, b$$

이다. 이것도 일반적으로 잘 정의할 수 있는데, 대충 다음과 같다.

- $\alpha = X^{\alpha_1} + X^{\alpha_2} + \dots + X^{\alpha_n}$ 일 때,

$$\alpha \&b[p] = X^{\alpha_1} \&b[p](\alpha_1) \dots (\alpha_{n-1}) X^{\alpha_n} \&b[p]$$

- $1 \& b[p] = b$
- $X^{\alpha+1} \& b[p] = \underbrace{X^{\alpha} \& b[p](\alpha) \cdots (\alpha) X^{\alpha} \& b[p]}_p$
- 나머지 모든 경우, $X^{\alpha} \& b[p] = X^{\alpha[p]} \& b[p]$

여기서 (α) 나 $\alpha[p]$ 는 역시 귀납적으로 적당히 정의된 ‘구조에 대응되는 쉼표’와 ‘극한 경우 p 번째 앞부분’ 정도이다.

그러면 BEAF의 정의는 다음과 같다. 첫 번째 칸의 수를 b , 두 번째 칸의 수를 p , 그 뒤로 처음으로 1이 아닌 칸의 수를 c , c 앞의 구조를 α , 그 뒤의 배열 전체를 $\#$ 이라 하자.

- $p = 1$ 이면 배열은 b 로 계산된다.
- c 가 존재하지 않으면, 즉 배열에 b 와 p 뿐이면 이는 b^p 으로 계산된다.
- 나머지 모든 경우,
 - c 의 앞쪽을 $\alpha \& b[p]$ 로 채운다.
 - c 가 그 행의 첫 번째 칸이 아닌 경우 (즉 $\alpha = \beta + 1$ 인 경우), c 바로 앞의 칸을 ‘전체 배열에서 p 를 1 줄인 배열의 계산 결과’로 바꾼다.
 - c 를 1 줄인다.

다른 표기법

BEAF가 단순하고 강력한 표기법임은 사실이지만 정의상 오류가 있기에 많이 사용하지는 않는다. 배열 형식의 표기법은 주로 BAN을 사용하는 추세이다. 것보다는 계속 대충 언급한 ‘서수를 통한 재귀 구조 단계’를 가장 많이 사용한다. 이는 사실 fgh라고 하는 다른 큰 수 표기법인데, 가장 단순하면서도 강력한 체계이다. 다만 서수라는 이론적 배경이 필요했기에 언급을 자제했다.

그 외에도 지수를 확장한 E 표기법이나 팩토리얼을 확장한 HAN 등이 있다. 그래도 역시 fgh 쓰는 게 가장 편하다. 그러나 fgh의 문제는 위로 갈수록 서수 표기법이 부족해진다는 것이다. 성장률이 너무 큰 나머지 아직 연구 중에 있는 바시쿠 행렬 체계라는 것도 있다.