無限リスト 胡振江

```
無限リストの例

- [1..] → [1,2,3,4,5,...]
- take n [1..] → [1..n]
- [m..] !! n → m+n
- map factorial [0..] → scan (*) 1 [1..]
- [x^2 | x <- [1..], odd x] → [1,9,25,...]
- [[m^n | m <-[1..]] | n<-[2..]]
→ [[1,4,9,16,...],
[1,8,27,64,...],...]
- [x^2 | x <- [1..], x^2 < 10]
→ [1,4,9]

1:4:9:1
```

練習問題

cubesとelemを次のように定義する。

cubes = [i^3 | i <- [1..]] x `elem` [] = False

 $x \cdot elem \cdot (y:ys) \mid x==y = True$

1. 64 'elem' cubes ==> ?

2. 65 'elem' cubes ==> ?

ф

反復

• f^n の定義

power f 0 = idpower f (n+1) = f. power f n

• 関数iterate

iterate f x = [x, f x, f^2 x, ...] iterate f x = x : iterate f (f x) 例:iterate (+1) 1 = [1,2,3,4,...] iterate (*2) 1 = [1,2,4,8,...] Iterate ('div' 10) 2718 = [2718, 271, 27, 2, 0, 0, ...]

[m..] = iterate (+1) m [m..n] = takewhile (<=n) (iterate (+1) m)

無限リストの利用例1

• 正の整数の数字を取り出す digits 2718 → [2,7,1,8]

> digits = reverse . [2,7,1,8] map ('mod` 10) . [8,1,7,2] takeWhile (/=0) . [2718,271,27,2] iterate ('div` 10) [2718,271,27,2,0,...]

L

無限リスト利用例2

 リストを長さnの部分に分割する group 2 [1,2,3,4,5,6] → [[1,2],[3,4],[5,6]]

group n = map (take n) .

takeWhile (/=[]) .

iterate (drop n)

42

素数の生成

ギリシャの数学者Eratosthenesの手法

- 1 数の並び2,3,...を書き下ろす
- 2 この並びの最初の要素pを素数として登録する
- 3 この並びからpの倍数を消去する
- 4 2へ戻る

Le

2 3 <u>4</u> 5 <u>6</u> 7 <u>8</u> 9 <u>10</u> 11 <u>12</u> 13 <u>14</u> 15 ... 3 5 7 <u>9</u> 11 13 <u>15</u> ... 5 7 11 13 ... primes = map head (iterate sieve [2..]) sieve (p:xs) = [x | x<-xs, x `mod` p /= 0]

極限としての無限リスト • 極限(limit) - 数学において無限の対象を扱うひとつの方法 - 例:π = 3.14159265358979323846...は 3 3.1 3.14 3.141 3.1415

4

の極限値であると考えられる。

無限リスト 「近似リスト」の列の極限とみなす 例:[1..]は次の列の一つの極限である 1:1 1:2:1 1:2:3:1 … 擬リスト (partial list): 値上で終るリスト x1: x2: …: xn: ⊥


```
• map (*2) [1..]の計算

map (*2) ⊥ = ⊥
map (*2) (1: ⊥) = 2: ⊥
map (*2) (1: 2: ⊥) = 2: 4: ⊥
...

→ [2,4,6,8,10,...]
```

```
• filter even [1..] の計算

filter even (1: ⊥) = ⊥
filter even (1: 2: ⊥) = 2: ⊥
filter even (1:2:3: ⊥) = 2: ⊥
filter even (1:2:3:4: ⊥) = 2:4: ⊥
...

→ [2,4,6,...]
```

```
練習問題

• filter (<10) (map (*2) [1..])

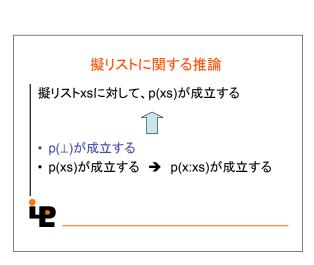
→ 2:4:6:8: ⊥

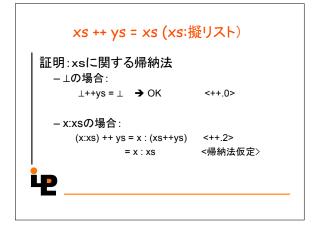
- Why?

• takeWhile (<10) (map (*2) [1..])

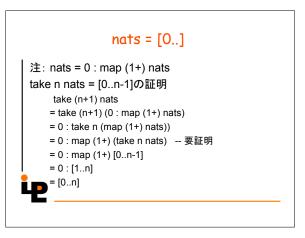
→ 2:4:6:8:[]

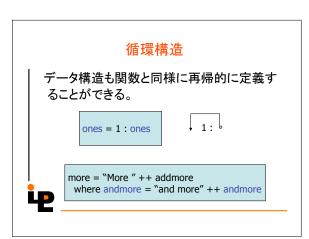
- Why?
```

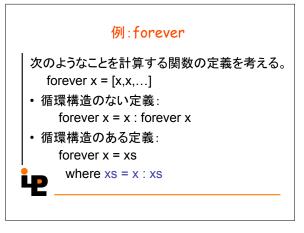












循環構造による効率化 iterative の二つの定義: iterate1 f x = x : map f (iterate1 f x) iterate2 f x = zs where zs = x : map f zs 最初のn項の計算では iterate1: O(n^2) iterate2: O(n)

大のような無限リストを生成する プログラムを設計せよ。 1. リストは重複のない上昇列である。 2. リストは1から始まる。 3. リストに数 x が含まれているならば、数 2*x, 3*x, 5*xもまたリストに含まれている。 4. リストにはそれ以外の数がふくまれていない。

hamming = 1: merge3 (map (2*) hamming) (map (3*) hamming) (map (5*) hamming) (map (5*) hamming) merge3 x y z = merge (x (merge y z)) merge (x:xs) (y:ys) | x==y = x : merge xs ys | x<y = x : merge xs (y:ys) | y<x = y : merge (x:xs) ys

期末試験

• 日時: 2月6日 8:30 - 10:00

• 場所: 63号室

・ 教科書、ノートを持ち込み可

• 内容:教科書1-7章

頑張ってください.

ie