Package 'Rfractran'

October 12, 2022

Type Package

Title A 'FRACTRAN' Interpreter and Some Helper Functions

Version 1.0

Date 2020-06-09

Author Carl Witthoft

Maintainer Carl Witthoft <carl@witthoft.com>

Depends gmp

Description 'FRACTRAN' is an obscure yet tantalizing programming language invented by John Conway of 'Game of Life' fame. The code consists of a sequence of fractions. The rules are simple. First, select an integer to initialize the process. Second, multiply the integer by the first fraction. If an integer results, start again with the new integer. If not, try the next fraction. Finally, if no such multiplication yields an integer, terminate the program. For more information, see <https://en.wikipedia.org/wiki/FRACTRAN> .

License LGPL-3

NeedsCompilation no

Repository CRAN

Date/Publication 2020-06-25 16:20:11 UTC

R topics documented:

`Rfractran-package` *A 'FRACTRAN' Interpreter and Some Helper Functions*

Description

'FRACTRAN' is an obscure yet tantalizing programming language invented by John Conway of 'Game of Life' fame. The code consists of a sequence of fractions. The rules are simple. First, select an integer to initialize the process. Second, multiply the integer by the first fraction. If an integer results, start again with the new integer. If not, try the next fraction. Finally, if no such multiplication yields an integer, terminate the program. For more information, see <https://en.wikipedia.org/wiki/FRACTRAN> .

Details

The DESCRIPTION file:

Package:	Rfractran
Type:	Package
Title:	A 'FRACTRAN' Interpreter and Some Helper Functions
Version:	1.0
Date:	2020-06-09
Author:	Carl Witthoft
Maintainer:	Carl Witthoft <carl@witthoft.com>
Depends:	gmp
Description:	'FRACTRAN' is an obscure yet tantalizing programming language invented by John Conway of 'Game of Life
License:	LGPL-3

FRACTRAN is an obscure yet tantalizing programming "language" invented by John Conway of "Game of Life" fame. The code consists of a sequence of fractions. The operation is simple. 1 - Initalize with an integer 2 - Multiply the integer by the first fraction. If an integer results, start again with the new integer. If not, try the next fraction. 3 - If no such multiplication yields an integer, terminate the program. One warning: there is a FRACTRAN program that can be found on the web which is supposed to generate the digits of pi. Unfortunately, it's known to have a bug which causes it not to work. So far nobody has found a correction for it.

Author(s)

Carl Witthoft Maintainer: Carl Witthoft <carl@witthoft.com>

References

https://esolangs.org/wiki/Fractran https://oeis.org/wiki/List_of_FRACTRAN_programs_to_compute_core_sequences

Examples

```
##Not Run
# addition:  { 3/2 } . Enter 2^a*3^b and get 3^(a+b) answer when terminated

# multiplication:
# { 455/33, 11/13, 1/11,  3/7, 11/2, 1/3 }
# then  enter 2^a*3^b  Terminates with  5^(a*b)

# prime generator (This function never terminates.)
#start with 10, feed to [7/3 99/98 13/49 39/35 36/91 10/143 49/13 7/11 1/2 91/1]
# and whenever the result of a run is of the form 10^p , p is prime
```

`fracAns` *A Q&D "Logulator" to Find Numbers of Interest From* `fracDo` *Output*

Description

In nearly all cases FRACTRAN's "result" is the exponent of some number. This function takes the logarithm of the specified base and identifies values which are integers (or nearly so, to the specified precision).

Usage

```
fracAns(intvec, logbase, logprec = 1e-05)
```

Arguments

`intvec`	A vector of `bigq` values, as returned from `fracDo`
`logbase`	The base of the desired logarithm, e.g. 2 or 10 in many cases.
`logprec`	A reasonably small value used to check whether the returned double should be considered to be an integer (thus ignoring binary precision errors)

Value

A vector of the integer values found

Author(s)

Carl Witthoft <carl@witthoft.com>

Examples

```
##---The prime generator doesn't terminate, so look for values.
#start with 10,  and result includes a series of values 10^p , p prime
fracp10 <- fracMake(c(7,99,13,39,36,10,49,7,1,91), c(3,98,49,35,91,143,13,11,2,1))
p10 <-fracDo(10,fractions = fracp10, tries = 1000)
foundp10 <-fracAns(p10,logbase = 10)
# [1] 1 2 3 5 7 # sorry about the "1" :-)
```

`fracDo` *The FRACTRAN Interpreter Function*

Description

This implements the FRACTRAN process, to wit For the first fraction f in the list for which nf is an integer, replace n by nf. Repeat this rule until no fraction in the list produces an integer when multiplied by n, then halt.

Usage

```
fracDo(input, fractions = NULL, nums, denoms, tries = 100,
 stopFun = NULL, liveUpdate = TRUE, ...)
```

Arguments

`input`	The starting integer for the given FRACTRAN code (fraction sequence)
`fractions`	The sequence of fractions. Must be in `bigq` form. If NULL, then `nums` and `denoms` must be supplied.
`nums`	Vector of integers (or `bigz` values) representing the numerators of the FRACTRAN fractions. If `fractions` is supplied, this is ignored.
`denoms`	Vector of integers (or `bigz` values) representing the denominators of the FRACTRAN fractions. If `fractions` is supplied, this is ignored.
`tries`	A 'safety' limiter on the number of times to process the next integer generated. This avoids possible infinite runs or other time-wasters.
`stopFun`	Optional user-supplied function that can be used to terminate a FRACTRAN run early, or to take min-run actions such as sending information to the console. See Details for more information.
`liveUpdate`	If set to TRUE, a few dots and words are sent to the console to indicate that the algorithm is still running.
`...`	Possible additional arguments for `stopFun` or future use.

Details

Some FRACTRAN programs do not terminate, most famously the prime generators. If a specific value is being looked for, an appropriate `stopFun` can be supplied to check for that value. `stopFun` must return a single logical value (R, as always will convert numerics to logical if necessary) indicating success as TRUE. The first argument to `stopFun` must accept a single `bigq` value. If there are more arguments, they must be entered after all named arguments. Note that this function does not have to send TRUE; it could be used solely to execute other commands mid-run. See `stopPrime` for one such example.

Value

A vector of all the `bigq` values generated. These all have denominator == 1, as they are the integers found in each iteration of the FRACTRAN algorithm.

Author(s)

Carl Witthoft <carl@witthoft.com

References

https://esolangs.org/wiki/Fractran https://oeis.org/wiki/List_of_FRACTRAN_programs_to_compute_core_sequences

See Also

fracMake

Examples

```
# addition:  { 3/2   } .  enter 2^a*3^b and get 3^(a+b) answer when terminated
addit <- fracDo(2^5*3^8,nums = 3, denoms = 2)
# Last value returned is what we want. But can't take log(bigq)
log(numerator(addit[length(addit)]),3)
```

---

fracMake *Function to Create a Sequence of* bigq *Fractions*

Description

Feed this function a collection of numerator and denominator values; get a vector of bigq fractions suitable for use as a FRACTRAN program.

Usage

```
fracMake(n, d = NULL)
```

Arguments

n	Vector of integers to be used as numerators, or a Nx2 array of integers where each row contains a num, denom pair. If this is an arry, the input d is ignored.
d	Vector of integers to be used as denominators. Ignored if n is an array

Value

Vector of bigq fractions.

Author(s)

Carl Witthoft <carl@witthoft.com>

stopPrime *Function to Stop* fracDo *When Finding Primes*

Description

This function is designed to be fed to fracDo as the argument stopFun. It will send a note to the console every time a prime number is found. It also serves as a sample of what a "stopping function" could look like - notice that despite the name, this function doesn't force-stop fracDo.

Usage

```
stopPrime(x, mantissa,returnVal = 0)
```

Arguments

x	The current integer generated with the FRACTRAN generator (fracDo.)
mantissa	Choose the mantissa based on the prime-generating algorithm selected. Must be an added argument to fracDo. See Details.
returnVal	Normally set to zero so as not to terminate the FRACTRAN program. If desired, set to one (or TRUE) to terminate after the first prime is found. Must be an added argument to fracDo. See Details.

Details

There are two well-known prime generators for FRACTRAN, one of which uses powers of two and the other uses powers of 10. Select the mantissa to match one of these, or potentially any other FRACTRAN program which returns values as (mantissa)^(result) . When running fracDo, you must enter the value as an argument after all named arguments. Similarly, if desired, enter a value for returnVal to fracDo.

Value

Returns the specified value of returnVal so that fracDo will or will not terminate. Its "useful" output is cat-ed to the console.

Author(s)

Carl Witthoft <carl@witthoft.com>

Examples

```
##---The prime generator doesn't terminate, so look for values.
#start with 10,  and result includes a series of values 10^p , p prime
fracp10 <- fracMake(c(7,99,13,39,36,10,49,7,1,91), c(3,98,49,35,91,143,13,11,2,1))
p10 <-fracDo(10,fractions = fracp10, tries = 1000,stopFun=stopPrime, mantissa = 10 )
# primes found are sent to console
# p10 contains all integers found.
```

Index