continuedFractions

Jon Sporring

October 15, 2019

1 Lærervejledningn

Emne Rekursion

Sværhedsgrad Middel

2 Introduktion

I denne opgave skal I regne med kædebrøker (continued fractions)[1]. Kædebrøker er lister af heltal, som repræsenterer reelle tal. Listen er endelig for rationelle tal og uendelig for irrationelle tal.

Kædebrøk til decimalbrøk En kædebrøk skrives som: $x = [q_0; q_1, q_2, \ldots]$, hvilket svarer til tallet,

$$x = q_0 + \frac{1}{q_1 + \frac{1}{q_2 + \ldots}}. \tag{1}$$

F.eks. svarer kædebrøken $[3; 4, 12, 4]$ til følgende decimaltal:

$$\begin{aligned} x &= 3 + \frac{1}{4 + \frac{1}{12 + \frac{1}{4}}} && (2) \\ &= 3 + \frac{1}{4 + \frac{1}{12.25}} && (3) \\ &= 3 + \frac{1}{4.081632653} && (4) \\ &= 3.245. && (5) \end{aligned}$$

Altså er $[3; 4, 12, 4] = 3.245$.

[1] https://en.wikipedia.org/wiki/Continued_fraction

Decimaltal til kædebrøk For et givet tal x på decimalform kan dets kædebrøk $[q_0; q_1, q_2, \ldots]$ udregnes ved følgende algoritme: Lad $x_0 = x$ og $i \geq 0$, udregn

$$q_i = \lfloor x_i \rfloor \tag{6}$$
$$r_i = x_i - q_i \tag{7}$$
$$x_{i+1} = 1/r_i \tag{8}$$
$$\tag{9}$$

indtil $r_i = 0$. F.eks. for decimaltallet $x = 3.245$ beregnes:

i	x_i	$q_i = \lfloor x_i \rfloor$	$r_i = x_i - q_i$	$x_{i+1} = 1/r_i$
0	3.245	3	0.245	4.081632653...
1	4.081632653...	4	0.081632653	12.25
2	12.25	12	0.25	4
3	4	4	0	-

Resultatet aflæses i anden søjle: $3.245 = [3; 4, 12, 4]$.

Heltalsbrøk til kædebrøk Kædebrøken for en heltals brøk t/n udregnes ved følgende algoritme: Lad $r_{-2} = t$ og $r_{-1} = n$ og $i \geq -2$, udregn

$$r_i = r_{i-2} \mathbin{\%} r_{i-1} \quad \text{(rest ved heltalsdivision)}, \tag{10}$$
$$q_i = r_{i-2} \operatorname{div} r_{i-1} \quad \text{(heltalsdivision)}, \tag{11}$$

indtil $r_{i-1} = 0$. Så vil $[q_0; q_1, \ldots, q_j]$ vil være t/n som kædebrøk. F.eks. for brøken $t/n = 649/200$ beregnes:

i	r_{i-2}	r_{i-1}	$r_i = r_{i-2} \mathbin{\%} r_{i-1}$	$q_i = r_{i-2} \operatorname{div} r_{i-1}$
0	649	200	49	3
1	200	49	4	4
2	49	4	1	12
3	4	1	0	4
4	1	0	-	-

Kædebrøken aflæses som højre søjle: $649/200 = [3; 4, 12, 4]$.

Kædebrøker af heltalsbrøker t/n er særligt effektive at udregne vha. Euclids algoritme for største fællesnævner. Algoritmen i Opgave **??** regner rekursivt på relationen mellem heltalsdivision og rest: Hvis $a = t \operatorname{div} n$ er heltalsdivision mellem t og n, og $b = t \mathbin{\%} n$ er resten efter heltalsdivision, så er $t = an + b$. For (10)–(11) skal der altså gælde, at $r_{i-2} = q_i r_{i-1} + r_i$. Algoritmen i Opgave **??** regner udelukkende på r_i som transformationen $(r_{i-2}, r_{i-1}) \to (r_{i-1}, r_i) = (r_{i-1}, r_{i-2} \mathbin{\%} r_{-1})$ indtil $(r_{i-2}, r_{i-1}) = (r_{i-2}, 0)$. Hvis man tilføjer beregning af q_i i rekursionen, har man samtidigt beregnet brøken som kædebrøk.

Kædebrøk til Heltalsbrøk(er) En kædebrøk kan approximeres som en heltalsbrøk $\frac{t_i}{n_i}, i \geq 0$ ved følgende algorime. Lad $t_{-2} = n_{-1} = 0$ og $t_{-1} = n_{-2} = 1$, udregn

$$t_i = q_i t_{i-1} + t_{i-2}, \tag{12}$$
$$n_i = q_i n_{i-1} + n_{i-2}, \tag{13}$$

indtil i er passende stor, eller der ikke er flere cifre q_i. F.eks. for kædebrøken $[3;4,12,4]$ beregnes,

$$\frac{t_0}{n_0} = \frac{q_0 t_{-1} + t_{-2}}{q_0 n_{-1} + n_{-2}} = \frac{3 \cdot 1 + 0}{3 \cdot 0 + 1} = \frac{3}{1} = 3, \tag{14}$$

$$\frac{t_1}{n_1} = \frac{q_1 t_0 + t_{-1}}{q_1 n_0 + n_{-1}} = \frac{4 \cdot 3 + 1}{4 \cdot 1 + 0} = \frac{13}{4} = 3.25, \tag{15}$$

$$\frac{t_2}{n_2} = \frac{q_2 t_1 + t_0}{q_2 n_1 + n_0} = \frac{12 \cdot 13 + 3}{12 \cdot 4 + 1} = \frac{159}{49} = 3.244897959\ldots, \tag{16}$$

$$\frac{t_3}{n_3} = \frac{q_3 t_2 + t_1}{q_3 n_2 + n_1} = \frac{4 \cdot 159 + 13}{4 \cdot 49 + 4} = \frac{649}{200} = 3.245. \tag{17}$$

Altså kan kædebrøkken $[3;4,12,4]$ approximeres som heltalsbrøkkerne $3/1$, $13/4$, $159/49$ og $649/200$ med gradvist stigende nøjagtighed.

3 Opgave(r)

1. Skriv en rekursiv funktion

```
cfrac2float : lst:int list -> float
```

som tager en liste af heltal som kædebrøk og udregner det tilsvarende reelle tal.

2. Skriv en rekursiv funktion

```
float2cfrac : x:float -> int list
```

som tager et reelt tal og udregner dens repræsentation som kædebrøk.

3. Skriv en rekursiv funktion

```
frac2cfrac : t:int -> n:int -> int list
```

som tager tæller og nævner i brøken t/n og udregner dens repræsentation som kædebrøk udelukkende ved brug af heltalstyper.

4. Skriv en rekursiv funktion

```
cfrac2frac : lst:int list -> i:int -> int * int
```

som tager en kædebrøk og et index og returnerer t_i/n_i approximationen som tuplen `(ti, ni)`.

5. Saml alle ovenstående funktioner i et bibliotek bestående af dets interface og implementationsfil (`continuedFraction.fsi` `continuedFraction.fs`), og lav en applikationsfil, der udfører en white- og black-box test af funktionerne.