

Continued Fractions

Manuel Eberl

March 17, 2025

Abstract

This article provides a formalisation of continued fractions of real numbers and their basic properties. It also contains a proof of the classic result that the irrational numbers with periodic continued fraction expansions are precisely the quadratic irrationals, i. e. real numbers that fulfil a non-trivial quadratic equation $ax^2 + bx + c = 0$ with integer coefficients.

Particular attention is given to the continued fraction expansion of \sqrt{D} for a non-square natural number D . Basic results about the length and structure of its period are provided, along with an executable algorithm to compute the period (and from it, the entire expansion).

This is then also used to provide a fairly efficient, executable, and fully formalised algorithm to compute solutions to Pell's equation $x^2 - Dy^2 = 1$. The performance is sufficiently good to find the solution to Archimedes's cattle problem in less than a second on a typical computer. This involves the value $D = 410286423278424$, for which the solution has over 200000 decimals.

Lastly, a derivation of the continued fraction expansions of Euler's number e and an executable function to compute continued fraction expansions using interval arithmetic is also provided.

Contents

1	Continued Fractions	3
1.1	Auxiliary results	3
1.2	Bounds on alternating decreasing sums	5
1.3	Non-canonical continued fractions	22
1.4	Approximation properties	27
1.5	Efficient code for convergents	30
1.6	Computing the continued fraction expansion of a rational number	31
2	Quadratic Irrationals	31
2.1	Basic results on rationality of square roots	32
2.2	Definition of quadratic irrationals	33
2.3	Real solutions of quadratic equations	34
2.4	Periodic continued fractions and quadratic irrationals	35
3	The continued fraction expansion of e	37
4	Continued fraction expansions for square roots of naturals	40
5	Lifting solutions of Pell's Equation	46
5.1	Auxiliary material	46
5.2	The lifting mechanism	48
5.3	Accelerated computation of the fundamental solution for non-squarefree inputs	49
6	The Connection between the continued fraction expansion of square roots and Pell's equation	50
7	Tests for Continued Fractions of Square Roots and Pell's Equation	52
7.1	Fundamental solutions of Pell's equation	53
7.2	Tests for other operations	54
8	Computing continued fraction expansions through interval arithmetic	54

1 Continued Fractions

```
theory Continued-Fractions
imports
  Complex-Main
  Coinductive.Lazy-LList
  Coinductive.Coinductive-Nat
  HOL-Number-Theory.Fib
  HOL-Library.BNF-Corec
  Coinductive.Coinductive-Stream
begin
```

1.1 Auxiliary results

```
coinductive linfinite :: 'a llist  $\Rightarrow$  bool where
  linfinite xs  $\Longrightarrow$  linfinite (LCons x xs)
```

```
lemma llength-llist-of-stream [simp]: llength (lstream xs) =  $\infty$ 
  <proof>
```

```
lemma linfinite-conv-llength: linfinite xs  $\longleftrightarrow$  llength xs =  $\infty$ 
  <proof>
```

```
definition lnth-default :: 'a  $\Rightarrow$  'a llist  $\Rightarrow$  nat  $\Rightarrow$  'a where
  lnth-default dft xs n = (if n < llength xs then lnth xs n else dft)
```

```
lemma lnth-default-code [code]:
  lnth-default dft xs n =
    (if lnull xs then dft else if n = 0 then lhd xs else lnth-default dft (ltl xs) (n -
    1))
  <proof>
```

```
lemma enat-le-iff:
  enat n  $\leq$  m  $\longleftrightarrow$  m =  $\infty$   $\vee$  ( $\exists$  m'. m = enat m'  $\wedge$  n  $\leq$  m')
  <proof>
```

```
lemma enat-less-iff:
  enat n < m  $\longleftrightarrow$  m =  $\infty$   $\vee$  ( $\exists$  m'. m = enat m'  $\wedge$  n < m')
  <proof>
```

```
lemma real-of-int-divide-in-Ints-iff:
  real-of-int a / real-of-int b  $\in$   $\mathbb{Z}$   $\longleftrightarrow$  b dvd a  $\vee$  b = 0
  <proof>
```

```
lemma frac-add-of-nat: frac (of-nat y + x) = frac x
  <proof>
```

```
lemma frac-add-of-int: frac (of-int y + x) = frac x
  <proof>
```

lemma *frac-fraction*: $\text{frac} (\text{real-of-int } a / \text{real-of-int } b) = (a \text{ mod } b) / b$
(proof)

lemma *Suc-fib-ge*: $\text{Suc} (\text{fib } n) \geq n$
(proof)

lemma *fib-ge*: $\text{fib } n \geq n - 1$
(proof)

lemma *frac-diff-of-nat-right [simp]*: $\text{frac} (x - \text{of-nat } y) = \text{frac } x$
(proof)

lemma *of-nat-ge-1-iff*: $\text{of-nat } n \geq (1 :: 'a :: \text{linordered-semidom}) \longleftrightarrow n > 0$
(proof)

lemma *not-frac-less-0*: $\neg \text{frac } x < 0$
(proof)

lemma *frac-le-1*: $\text{frac } x \leq 1$
(proof)

lemma *divide-in-Rats-iff1*:
 $(x::\text{real}) \in \mathbb{Q} \implies x \neq 0 \implies x / y \in \mathbb{Q} \longleftrightarrow y \in \mathbb{Q}$
(proof)

lemma *divide-in-Rats-iff2*:
 $(y::\text{real}) \in \mathbb{Q} \implies y \neq 0 \implies x / y \in \mathbb{Q} \longleftrightarrow x \in \mathbb{Q}$
(proof)

lemma *add-in-Rats-iff1*: $x \in \mathbb{Q} \implies x + y \in \mathbb{Q} \longleftrightarrow y \in \mathbb{Q}$
(proof)

lemma *add-in-Rats-iff2*: $y \in \mathbb{Q} \implies x + y \in \mathbb{Q} \longleftrightarrow x \in \mathbb{Q}$
(proof)

lemma *diff-in-Rats-iff1*: $x \in \mathbb{Q} \implies x - y \in \mathbb{Q} \longleftrightarrow y \in \mathbb{Q}$
(proof)

lemma *diff-in-Rats-iff2*: $y \in \mathbb{Q} \implies x - y \in \mathbb{Q} \longleftrightarrow x \in \mathbb{Q}$
(proof)

lemma *frac-in-Rats-iff [simp]*: $\text{frac } x \in \mathbb{Q} \longleftrightarrow x \in \mathbb{Q}$
(proof)

lemma *filterlim-sequentially-shift*:
 $\text{filterlim} (\lambda n. f (n + m)) F \text{ sequentially} \longleftrightarrow \text{filterlim } f F \text{ sequentially}$
(proof)

1.2 Bounds on alternating decreasing sums

lemma *alternating-decreasing-sum-bounds*:

fixes $f :: \text{nat} \Rightarrow 'a :: \{\text{linordered-ring}, \text{ring-1}\}$
assumes $m \leq n \wedge k. k \in \{m..n\} \implies f k \geq 0$
 $\wedge k. k \in \{m..<n\} \implies f (\text{Suc } k) \leq f k$
defines $S \equiv (\lambda m. (\sum k=m..n. (-1) ^ k * f k))$
shows *if even m then S m ∈ {0..f m} else S m ∈ {-f m..0}*
<proof>

lemma *alternating-decreasing-sum-bounds'*:

fixes $f :: \text{nat} \Rightarrow 'a :: \{\text{linordered-ring}, \text{ring-1}\}$
assumes $m < n \wedge k. k \in \{m..n-1\} \implies f k \geq 0$
 $\wedge k. k \in \{m..<n-1\} \implies f (\text{Suc } k) \leq f k$
defines $S \equiv (\lambda m. (\sum k=m..<n. (-1) ^ k * f k))$
shows *if even m then S m ∈ {0..f m} else S m ∈ {-f m..0}*
<proof>

lemma *alternating-decreasing-sum-upper-bound*:

fixes $f :: \text{nat} \Rightarrow 'a :: \{\text{linordered-ring}, \text{ring-1}\}$
assumes $m \leq n \wedge k. k \in \{m..n\} \implies f k \geq 0$
 $\wedge k. k \in \{m..<n\} \implies f (\text{Suc } k) \leq f k$
shows $(\sum k=m..n. (-1) ^ k * f k) \leq f m$
<proof>

lemma *alternating-decreasing-sum-upper-bound'*:

fixes $f :: \text{nat} \Rightarrow 'a :: \{\text{linordered-ring}, \text{ring-1}\}$
assumes $m < n \wedge k. k \in \{m..n-1\} \implies f k \geq 0$
 $\wedge k. k \in \{m..<n-1\} \implies f (\text{Suc } k) \leq f k$
shows $(\sum k=m..<n. (-1) ^ k * f k) \leq f m$
<proof>

lemma *abs-alternating-decreasing-sum-upper-bound*:

fixes $f :: \text{nat} \Rightarrow 'a :: \{\text{linordered-ring}, \text{ring-1}\}$
assumes $m \leq n \wedge k. k \in \{m..n\} \implies f k \geq 0$
 $\wedge k. k \in \{m..<n\} \implies f (\text{Suc } k) \leq f k$
shows $|(\sum k=m..n. (-1) ^ k * f k)| \leq f m$ (**is abs ?S ≤ -**)
<proof>

lemma *abs-alternating-decreasing-sum-upper-bound'*:

fixes $f :: \text{nat} \Rightarrow 'a :: \{\text{linordered-ring}, \text{ring-1}\}$
assumes $m < n \wedge k. k \in \{m..n-1\} \implies f k \geq 0$
 $\wedge k. k \in \{m..<n-1\} \implies f (\text{Suc } k) \leq f k$
shows $|(\sum k=m..<n. (-1) ^ k * f k)| \leq f m$
<proof>

lemma *abs-alternating-decreasing-sum-lower-bound*:

fixes $f :: \text{nat} \Rightarrow 'a :: \{\text{linordered-ring}, \text{ring-1}\}$
assumes $m < n \wedge k. k \in \{m..n\} \implies f k \geq 0$
 $\wedge k. k \in \{m..<n\} \implies f (\text{Suc } k) \leq f k$

shows $|(\sum_{k=m..n}. (-1) ^ k * f k)| \geq f m - f (Suc m)$
 <proof>

lemma *abs-alternating-decreasing-sum-lower-bound'*:
fixes $f :: nat \Rightarrow 'a :: \{linordered-ring, ring-1\}$
assumes $m+1 < n \wedge k. k \in \{m..n\} \implies f k \geq 0$
 $\wedge k. k \in \{m..<n\} \implies f (Suc k) \leq f k$
shows $|(\sum_{k=m..<n}. (-1) ^ k * f k)| \geq f m - f (Suc m)$
 <proof>

lemma *alternating-decreasing-suminf-bounds*:
assumes $\wedge k. f k \geq (0 :: real) \wedge k. f (Suc k) \leq f k$
 $f \longrightarrow 0$
shows $(\sum k. (-1) ^ k * f k) \in \{f 0 - f 1..f 0\}$
 <proof>

lemma
assumes $\wedge k. k \geq m \implies f k \geq (0 :: real)$
 $\wedge k. k \geq m \implies f (Suc k) \leq f k f \longrightarrow 0$
defines $S \equiv (\sum k. (-1) ^ (k + m) * f (k + m))$
shows *summable-alternating-decreasing*: *summable* $(\lambda k. (-1) ^ (k + m) * f (k + m))$
and *alternating-decreasing-suminf-bounds'*:
if even m then $S \in \{f m - f (Suc m) .. f m\}$
else $S \in \{-f m..f (Suc m) - f m\}$ (**is** ?th1)
and *abs-alternating-decreasing-suminf*:
 $abs S \in \{f m - f (Suc m)..f m\}$ (**is** ?th2)
 <proof>

lemma
assumes $\wedge k. k \geq m \implies f k \geq (0 :: real)$
 $\wedge k. k \geq m \implies f (Suc k) < f k f \longrightarrow 0$
defines $S \equiv (\sum k. (-1) ^ (k + m) * f (k + m))$
shows *alternating-decreasing-suminf-bounds-strict'*:
if even m then $S \in \{f m - f (Suc m) < .. < f m\}$
else $S \in \{-f m < .. < f (Suc m) - f m\}$ (**is** ?th1)
and *abs-alternating-decreasing-suminf-strict*:
 $abs S \in \{f m - f (Suc m) < .. < f m\}$ (**is** ?th2)
 <proof>

datatype *cfrac* = CFrac int nat llist

quickcheck-generator *cfrac* constructors: CFrac

lemma *type-definition-cfrac'*:
type-definition $(\lambda x. case x of CFrac a b \Rightarrow (a, b)) (\lambda(x,y). CFrac x y) UNIV$
 <proof>

setup-lifting *type-definition-cfrac'*

lift-definition *cfrac-of-int* :: *int* \Rightarrow *cfrac* **is**
 $\lambda n. (n, LNil) \langle proof \rangle$

lemma *cfrac-of-int-code* [*code*]: *cfrac-of-int* *n* = *CFrac* *n* *LNil*
 $\langle proof \rangle$

lift-definition *cfrac-of-stream* :: *int stream* \Rightarrow *cfrac* **is**
 $\lambda xs. (shd\ xs, llist-of-stream\ (smap\ (\lambda x. nat\ (x - 1))\ (stl\ xs))) \langle proof \rangle$

instantiation *cfrac* :: *zero*
begin
definition *zero-cfrac* **where** *0* = *cfrac-of-int* *0*
instance $\langle proof \rangle$
end

instantiation *cfrac* :: *one*
begin
definition *one-cfrac* **where** *1* = *cfrac-of-int* *1*
instance $\langle proof \rangle$
end

lift-definition *cfrac-tl* :: *cfrac* \Rightarrow *cfrac* **is**
 $\lambda(-, bs) \Rightarrow case\ bs\ of\ LNil \Rightarrow (1, LNil) \mid LCons\ b\ bs' \Rightarrow (int\ b + 1, bs') \langle proof \rangle$

lemma *cfrac-tl-code* [*code*]:
cfrac-tl (*CFrac* *a* *bs*) =
 $(case\ bs\ of\ LNil \Rightarrow CFrac\ 1\ LNil \mid LCons\ b\ bs' \Rightarrow CFrac\ (int\ b + 1)\ bs') \langle proof \rangle$

definition *cfrac-drop* :: *nat* \Rightarrow *cfrac* \Rightarrow *cfrac* **where**
cfrac-drop *n* *c* = (*cfrac-tl* $\overset{\sim}{\sim}$ *n*) *c*

lemma *cfrac-drop-Suc-right*: *cfrac-drop* (*Suc* *n*) *c* = *cfrac-drop* *n* (*cfrac-tl* *c*)
 $\langle proof \rangle$

lemma *cfrac-drop-Suc-left*: *cfrac-drop* (*Suc* *n*) *c* = *cfrac-tl* (*cfrac-drop* *n* *c*)
 $\langle proof \rangle$

lemma *cfrac-drop-add*: *cfrac-drop* (*m* + *n*) *c* = *cfrac-drop* *m* (*cfrac-drop* *n* *c*)
 $\langle proof \rangle$

lemma *cfrac-drop-0* [*simp*]: *cfrac-drop* *0* = ($\lambda x. x$)
 $\langle proof \rangle$

lemma *cfrac-drop-1* [*simp*]: *cfrac-drop* *1* = *cfrac-tl*
 $\langle proof \rangle$

lift-definition *cfrac-length* :: *cfrac* \Rightarrow *enat* **is**
 $\lambda(-, bs) \Rightarrow \text{llength } bs$ *<proof>*

lemma *cfrac-length-code* [*code*]: *cfrac-length* (*CFrac* *a* *bs*) = *llength* *bs*
<proof>

lemma *cfrac-length-tl* [*simp*]: *cfrac-length* (*cfrac-tl* *c*) = *cfrac-length* *c* - 1
<proof>

lemma *enat-diff-Suc-right* [*simp*]: *m* - *enat* (*Suc* *n*) = *m* - *n* - 1
<proof>

lemma *cfrac-length-drop* [*simp*]: *cfrac-length* (*cfrac-drop* *n* *c*) = *cfrac-length* *c* -
n
<proof>

lemma *cfrac-length-of-stream* [*simp*]: *cfrac-length* (*cfrac-of-stream* *xs*) = ∞
<proof>

lift-definition *cfrac-nth* :: *cfrac* \Rightarrow *nat* \Rightarrow *int* **is**
 $\lambda(a :: \text{int}, bs :: \text{nat llist}). \lambda(n :: \text{nat}).$
 if *n* = 0 *then* *a*
 else if *n* \leq *llength* *bs* *then* *int* (*lnth* *bs* (*n* - 1)) + 1 *else* 1 *<proof>*

lemma *cfrac-nth-code* [*code*]:
cfrac-nth (*CFrac* *a* *bs*) *n* = (*if* *n* = 0 *then* *a* *else* *lnth-default* 0 *bs* (*n* - 1) + 1)
<proof>

lemma *cfrac-nth-nonneg* [*simp, intro*]: *n* > 0 \implies *cfrac-nth* *c* *n* \geq 0
<proof>

lemma *cfrac-nth-nonzero* [*simp*]: *n* > 0 \implies *cfrac-nth* *c* *n* \neq 0
<proof>

lemma *cfrac-nth-pos*[*simp, intro*]: *n* > 0 \implies *cfrac-nth* *c* *n* > 0
<proof>

lemma *cfrac-nth-ge-1*[*simp, intro*]: *n* > 0 \implies *cfrac-nth* *c* *n* \geq 1
<proof>

lemma *cfrac-nth-not-less-1*[*simp, intro*]: *n* > 0 \implies \neg *cfrac-nth* *c* *n* < 1
<proof>

lemma *cfrac-nth-tl* [*simp*]: *cfrac-nth* (*cfrac-tl* *c*) *n* = *cfrac-nth* *c* (*Suc* *n*)
<proof>

lemma *cfrac-nth-drop* [*simp*]: *cfrac-nth* (*cfrac-drop* *n* *c*) *m* = *cfrac-nth* *c* (*m* + *n*)
<proof>

lemma *cfrac-nth-0-of-int* [simp]: $cfrac\text{-}nth\ (cfrac\text{-}of\text{-}int\ n)\ 0 = n$
⟨proof⟩

lemma *cfrac-nth-gt0-of-int* [simp]: $m > 0 \implies cfrac\text{-}nth\ (cfrac\text{-}of\text{-}int\ n)\ m = 1$
⟨proof⟩

lemma *cfrac-nth-of-stream*:
assumes $sset\ (stl\ xs) \subseteq \{0<..\}$
shows $cfrac\text{-}nth\ (cfrac\text{-}of\text{-}stream\ xs)\ n = snth\ xs\ n$
⟨proof⟩

lift-definition *cfrac* :: $(nat \Rightarrow int) \Rightarrow cfrac$ **is**
 $\lambda f.\ (f\ 0,\ inf\text{-}llist\ (\lambda n.\ nat\ (f\ (Suc\ n) - 1)))$ ⟨proof⟩

definition *is-cfrac* :: $(nat \Rightarrow int) \Rightarrow bool$ **where** $is\text{-}cfrac\ f \longleftrightarrow (\forall n > 0.\ f\ n > 0)$

lemma *cfrac-nth-cfrac* [simp]:
assumes *is-cfrac* *f*
shows $cfrac\text{-}nth\ (cfrac\ f)\ n = f\ n$
⟨proof⟩

lemma *llength-eq-infty-lnth*: $llength\ b = \infty \implies inf\text{-}llist\ (lnth\ b) = b$
⟨proof⟩

lemma *cfrac-cfrac-nth* [simp]: $cfrac\text{-}length\ c = \infty \implies cfrac\ (cfrac\text{-}nth\ c) = c$
⟨proof⟩

lemma *cfrac-length-cfrac* [simp]: $cfrac\text{-}length\ (cfrac\ f) = \infty$
⟨proof⟩

lift-definition *cfrac-of-list* :: $int\ list \Rightarrow cfrac$ **is**
 $\lambda xs.\ if\ xs = []\ then\ (0,\ LNil)\ else\ (hd\ xs,\ llist\text{-}of\ (map\ (\lambda n.\ nat\ n - 1)\ (tl\ xs)))$
⟨proof⟩

lemma *cfrac-length-of-list* [simp]: $cfrac\text{-}length\ (cfrac\text{-}of\text{-}list\ xs) = length\ xs - 1$
⟨proof⟩

lemma *cfrac-of-list-Nil* [simp]: $cfrac\text{-}of\text{-}list\ [] = 0$
⟨proof⟩

lemma *cfrac-nth-of-list* [simp]:
assumes $n < length\ xs$ **and** $\forall i \in \{0 < .. < length\ xs\}.\ xs\ !\ i > 0$
shows $cfrac\text{-}nth\ (cfrac\text{-}of\text{-}list\ xs)\ n = xs\ !\ n$
⟨proof⟩

primcorec *cfrac-of-real-aux* :: $real \Rightarrow nat\ llist$ **where**

cfrac-of-real-aux $x =$
 (if $x \in \{0 < .. < 1\}$ then $LCons$ (nat $\lfloor 1/x \rfloor - 1$) (*cfrac-of-real-aux* (frac (1/x)))
 else $LNil$)

lemma *cfrac-of-real-aux-code* [code]:

cfrac-of-real-aux $x =$
 (if $x > 0 \wedge x < 1$ then $LCons$ (nat $\lfloor 1/x \rfloor - 1$) (*cfrac-of-real-aux* (frac (1/x)))
 else $LNil$)
 ⟨proof⟩

lemma *cfrac-of-real-aux-LNil* [simp]: $x \notin \{0 < .. < 1\} \implies \text{cfrac-of-real-aux } x = LNil$
 ⟨proof⟩

lemma *cfrac-of-real-aux-0* [simp]: *cfrac-of-real-aux* $0 = LNil$
 ⟨proof⟩

lemma *cfrac-of-real-aux-eq-LNil-iff* [simp]: *cfrac-of-real-aux* $x = LNil \iff x \notin \{0 < .. < 1\}$
 ⟨proof⟩

lemma *lnth-cfrac-of-real-aux*:

assumes $n < \text{llength } (\text{cfrac-of-real-aux } x)$
shows $\text{lnth } (\text{cfrac-of-real-aux } x) (\text{Suc } n) = \text{lnth } (\text{cfrac-of-real-aux } (\text{frac } (1/x)))$
 n
 ⟨proof⟩

lift-definition *cfrac-of-real* :: *real* \implies *cfrac* **is**

$\lambda x. (\lfloor x \rfloor, \text{cfrac-of-real-aux } (\text{frac } x))$ ⟨proof⟩

lemma *cfrac-of-real-code* [code]: *cfrac-of-real* $x = CFrac \lfloor x \rfloor (\text{cfrac-of-real-aux } (\text{frac } x))$
 ⟨proof⟩

lemma *eq-epred-iff*: $m = \text{epred } n \iff m = 0 \wedge n = 0 \vee n = \text{eSuc } m$
 ⟨proof⟩

lemma *epred-eq-iff*: $\text{epred } n = m \iff m = 0 \wedge n = 0 \vee n = \text{eSuc } m$
 ⟨proof⟩

lemma *epred-less*: $n > 0 \implies n \neq \infty \implies \text{epred } n < n$
 ⟨proof⟩

lemma *cfrac-nth-of-real-0* [simp]:

cfrac-nth (*cfrac-of-real* x) $0 = \lfloor x \rfloor$
 ⟨proof⟩

lemma *frac-eq-0* [simp]: $x \in \mathbb{Z} \implies \text{frac } x = 0$
 ⟨proof⟩

lemma *cfrac-tl-of-real*:

assumes $x \notin \mathbb{Z}$

shows $cfrac\text{-tl} (cfrac\text{-of-real } x) = cfrac\text{-of-real } (1 / frac\ x)$

<proof>

lemma *cfrac-nth-of-real-Suc*:

assumes $x \notin \mathbb{Z}$

shows $cfrac\text{-nth} (cfrac\text{-of-real } x) (Suc\ n) = cfrac\text{-nth} (cfrac\text{-of-real } (1 / frac\ x))\ n$

<proof>

fun *conv* :: $cfrac \Rightarrow nat \Rightarrow real$ **where**

$conv\ c\ 0 = real\text{-of-int} (cfrac\text{-nth}\ c\ 0)$

| $conv\ c\ (Suc\ n) = real\text{-of-int} (cfrac\text{-nth}\ c\ 0) + 1 / conv\ (cfrac\text{-tl}\ c)\ n$

The numerator and denominator of a convergent:

fun *conv-num* :: $cfrac \Rightarrow nat \Rightarrow int$ **where**

$conv\text{-num}\ c\ 0 = cfrac\text{-nth}\ c\ 0$

| $conv\text{-num}\ c\ (Suc\ 0) = cfrac\text{-nth}\ c\ 1 * cfrac\text{-nth}\ c\ 0 + 1$

| $conv\text{-num}\ c\ (Suc\ (Suc\ n)) = cfrac\text{-nth}\ c\ (Suc\ (Suc\ n)) * conv\text{-num}\ c\ (Suc\ n) + conv\text{-num}\ c\ n$

fun *conv-denom* :: $cfrac \Rightarrow nat \Rightarrow int$ **where**

$conv\text{-denom}\ c\ 0 = 1$

| $conv\text{-denom}\ c\ (Suc\ 0) = cfrac\text{-nth}\ c\ 1$

| $conv\text{-denom}\ c\ (Suc\ (Suc\ n)) = cfrac\text{-nth}\ c\ (Suc\ (Suc\ n)) * conv\text{-denom}\ c\ (Suc\ n) + conv\text{-denom}\ c\ n$

lemma *conv-num-rec*:

$n \geq 2 \implies conv\text{-num}\ c\ n = cfrac\text{-nth}\ c\ n * conv\text{-num}\ c\ (n - 1) + conv\text{-num}\ c\ (n - 2)$

<proof>

lemma *conv-denom-rec*:

$n \geq 2 \implies conv\text{-denom}\ c\ n = cfrac\text{-nth}\ c\ n * conv\text{-denom}\ c\ (n - 1) + conv\text{-denom}\ c\ (n - 2)$

<proof>

fun *conv'* :: $cfrac \Rightarrow nat \Rightarrow real \Rightarrow real$ **where**

$conv'\ c\ 0\ z = z$

| $conv'\ c\ (Suc\ n)\ z = conv'\ c\ n\ (real\text{-of-int} (cfrac\text{-nth}\ c\ n) + 1 / z)$

Occasionally, it can be useful to extend the domain of *conv-num* and *conv-denom* to -1 and -2 .

definition *conv-num-int* :: $cfrac \Rightarrow int \Rightarrow int$ **where**

$conv\text{-}num\text{-}int\ c\ n = (if\ n = -1\ then\ 1\ else\ if\ n < 0\ then\ 0\ else\ conv\text{-}num\ c\ (nat\ n))$

definition $conv\text{-}denom\text{-}int :: cfrac \Rightarrow int \Rightarrow int$ **where**

$conv\text{-}denom\text{-}int\ c\ n = (if\ n = -2\ then\ 1\ else\ if\ n < 0\ then\ 0\ else\ conv\text{-}denom\ c\ (nat\ n))$

lemma $conv\text{-}num\text{-}int\text{-}rec$:

assumes $n \geq 0$

shows $conv\text{-}num\text{-}int\ c\ n = cfrac\text{-}nth\ c\ (nat\ n) * conv\text{-}num\text{-}int\ c\ (n - 1) + conv\text{-}num\text{-}int\ c\ (n - 2)$
 $\langle proof \rangle$

lemma $conv\text{-}denom\text{-}int\text{-}rec$:

assumes $n \geq 0$

shows $conv\text{-}denom\text{-}int\ c\ n = cfrac\text{-}nth\ c\ (nat\ n) * conv\text{-}denom\text{-}int\ c\ (n - 1) + conv\text{-}denom\text{-}int\ c\ (n - 2)$
 $\langle proof \rangle$

The number $[a_0; a_1, a_2, \dots]$ that the infinite continued fraction converges to:

definition $cfrac\text{-}lim :: cfrac \Rightarrow real$ **where**

$cfrac\text{-}lim\ c =$

$(case\ cfrac\text{-}length\ c\ of\ \infty \Rightarrow lim\ (conv\ c) \mid enat\ l \Rightarrow conv\ c\ l)$

lemma $cfrac\text{-}lim\text{-}code$ $[code]$:

$cfrac\text{-}lim\ c =$

$(case\ cfrac\text{-}length\ c\ of\ enat\ l \Rightarrow conv\ c\ l$

$\mid - \Rightarrow Code.abort\ (STR\ "Cannot\ compute\ infinite\ continued\ fraction")\ (\lambda.\ cfrac\text{-}lim\ c))$

$\langle proof \rangle$

definition $cfrac\text{-}remainder$ **where** $cfrac\text{-}remainder\ c\ n = cfrac\text{-}lim\ (cfrac\text{-}drop\ n\ c)$

lemmas $conv'\text{-}Suc\text{-}right = conv'.simps(2)$

lemma $conv'\text{-}Suc\text{-}left$:

assumes $z > 0$

shows $conv'\ c\ (Suc\ n)\ z =$

$real\text{-}of\text{-}int\ (cfrac\text{-}nth\ c\ 0) + 1 / conv'\ (cfrac\text{-}tl\ c)\ n\ z$

$\langle proof \rangle$

lemmas $[simp\ del] = conv'.simps(2)$

lemma $conv'\text{-}left\text{-}induct$:

assumes $\bigwedge c. P\ c\ 0\ z \wedge c\ n. P\ (cfrac\text{-}tl\ c)\ n\ z \implies P\ c\ (Suc\ n)\ z$

shows $P\ c\ n\ z$

$\langle proof \rangle$

lemma *enat-less-diff-conv* [*simp*]:

assumes $a = \infty \vee b < \infty \vee c < \infty$

shows $a < c - (b :: \text{enat}) \longleftrightarrow a + b < c$

<proof>

lemma *conv-eq-conv'*: $\text{conv } c \ n = \text{conv}' \ c \ n \ (\text{cfrac-nth } c \ n)$

<proof>

lemma *conv-num-pos'*:

assumes $\text{cfrac-nth } c \ 0 > 0$

shows $\text{conv-num } c \ n > 0$

<proof>

lemma *conv-num-nonneg*: $\text{cfrac-nth } c \ 0 \geq 0 \implies \text{conv-num } c \ n \geq 0$

<proof>

lemma *conv-num-pos*:

$\text{cfrac-nth } c \ 0 \geq 0 \implies n > 0 \implies \text{conv-num } c \ n > 0$

<proof>

lemma *conv-denom-pos* [*simp*, *intro*]: $\text{conv-denom } c \ n > 0$

<proof>

lemma *conv-denom-not-nonpos* [*simp*]: $\neg \text{conv-denom } c \ n \leq 0$

<proof>

lemma *conv-denom-not-neg* [*simp*]: $\neg \text{conv-denom } c \ n < 0$

<proof>

lemma *conv-denom-nonzero* [*simp*]: $\text{conv-denom } c \ n \neq 0$

<proof>

lemma *conv-denom-nonneg* [*simp*, *intro*]: $\text{conv-denom } c \ n \geq 0$

<proof>

lemma *conv-num-int-neg1* [*simp*]: $\text{conv-num-int } c \ (-1) = 1$

<proof>

lemma *conv-num-int-neg* [*simp*]: $n < 0 \implies n \neq -1 \implies \text{conv-num-int } c \ n = 0$

<proof>

lemma *conv-num-int-of-nat* [*simp*]: $\text{conv-num-int } c \ (\text{int } n) = \text{conv-num } c \ n$

<proof>

lemma *conv-num-int-nonneg* [*simp*]: $n \geq 0 \implies \text{conv-num-int } c \ n = \text{conv-num } c \ n$

(*nat n*)

<proof>

lemma *conv-denom-int-neg2* [simp]: $\text{conv-denom-int } c \ (-2) = 1$
⟨proof⟩

lemma *conv-denom-int-neg* [simp]: $n < 0 \implies n \neq -2 \implies \text{conv-denom-int } c \ n = 0$
⟨proof⟩

lemma *conv-denom-int-of-nat* [simp]: $\text{conv-denom-int } c \ (\text{int } n) = \text{conv-denom } c \ n$
⟨proof⟩

lemma *conv-denom-int-nonneg* [simp]: $n \geq 0 \implies \text{conv-denom-int } c \ n = \text{conv-denom } c \ (\text{nat } n)$
⟨proof⟩

lemmas *conv-Suc* [simp del] = *conv.simps*(2)

lemma *conv'-gt-1*:
 assumes *cfrac-nth* $c \ 0 > 0 \ x > 1$
 shows $\text{conv}' \ c \ n \ x > 1$
 ⟨proof⟩

lemma *enat-eq-iff*: $a = \text{enat } b \longleftrightarrow (\exists a'. a = \text{enat } a' \wedge a' = b)$
⟨proof⟩

lemma *eq-enat-iff*: $\text{enat } a = b \longleftrightarrow (\exists b'. b = \text{enat } b' \wedge a = b')$
⟨proof⟩

lemma *enat-diff-one* [simp]: $\text{enat } a - 1 = \text{enat } (a - 1)$
⟨proof⟩

lemma *conv'-eqD*:
 assumes $\text{conv}' \ c \ n \ x = \text{conv}' \ c' \ n \ x \ x > 1 \ m < n$
 shows $\text{cfrac-nth } c \ m = \text{cfrac-nth } c' \ m$
 ⟨proof⟩

context
 fixes $c :: \text{cfrac}$ **and** $h \ k$
 defines $h \equiv \text{conv-num } c$ **and** $k \equiv \text{conv-denom } c$
begin

lemma *conv'-num-denom-aux*:
 assumes $z: z > 0$
 shows $\text{conv}' \ c \ (\text{Suc } (\text{Suc } n)) \ z * (z * k \ (\text{Suc } n) + k \ n) =$
 $(z * h \ (\text{Suc } n) + h \ n)$
 ⟨proof⟩

lemma *conv'-num-denom*:
 assumes $z > 0$

shows $\text{conv}' c (\text{Suc } (\text{Suc } n)) z =$
 $(z * h (\text{Suc } n) + h n) / (z * k (\text{Suc } n) + k n)$

<proof>

lemma *conv-num-denom*: $\text{conv } c n = h n / k n$

<proof>

lemma *conv'-num-denom'*:

assumes $z > 0$ **and** $n \geq 2$

shows $\text{conv}' c n z = (z * h (n - 1) + h (n - 2)) / (z * k (n - 1) + k (n - 2))$

<proof>

lemma *conv'-num-denom-int*:

assumes $z > 0$

shows $\text{conv}' c n z =$

$(z * \text{conv-num-int } c (\text{int } n - 1) + \text{conv-num-int } c (\text{int } n - 2)) /$
 $(z * \text{conv-denom-int } c (\text{int } n - 1) + \text{conv-denom-int } c (\text{int } n - 2))$

<proof>

lemma *conv-nonneg*: $\text{cfraction } c 0 \geq 0 \implies \text{conv } c n \geq 0$

<proof>

lemma *conv-pos*:

assumes $\text{cfraction } c 0 > 0$

shows $\text{conv } c n > 0$

<proof>

lemma *conv-num-denom-prod-diff*:

$k n * h (\text{Suc } n) - k (\text{Suc } n) * h n = (-1) ^ n$

<proof>

lemma *conv-num-denom-prod-diff'*:

$k (\text{Suc } n) * h n - k n * h (\text{Suc } n) = (-1) ^ \text{Suc } n$

<proof>

lemma

fixes $n :: \text{int}$

assumes $n \geq -2$

shows *conv-num-denom-int-prod-diff*:

$\text{conv-denom-int } c n * \text{conv-num-int } c (n + 1) -$

$\text{conv-denom-int } c (n + 1) * \text{conv-num-int } c n = (-1) ^ (\text{nat } (n + 2))$

(is ?th1)

and *conv-num-denom-int-prod-diff'*:

$\text{conv-denom-int } c (n + 1) * \text{conv-num-int } c n -$

$\text{conv-denom-int } c n * \text{conv-num-int } c (n + 1) = (-1) ^ (\text{nat } (n + 3))$

(is ?th2)

<proof>

lemma *coprime-conv-num-denom*: $\text{coprime } (h \ n) \ (k \ n)$
<proof>

lemma *coprime-conv-num-denom-int*:
 assumes $n \geq -2$
 shows $\text{coprime } (\text{conv-num-int } c \ n) \ (\text{conv-denom-int } c \ n)$
<proof>

lemma *mono-conv-num*:
 assumes $\text{cfraction } c \ 0 \geq 0$
 shows $\text{mono } h$
<proof>

lemma *mono-conv-denom*: $\text{mono } k$
<proof>

lemma *conv-num-leI*: $\text{cfraction } c \ 0 \geq 0 \implies m \leq n \implies h \ m \leq h \ n$
<proof>

lemma *conv-denom-leI*: $m \leq n \implies k \ m \leq k \ n$
<proof>

lemma *conv-denom-lessI*:
 assumes $m < n \ 1 < n$
 shows $k \ m < k \ n$
<proof>

lemma *conv-num-lower-bound*:
 assumes $\text{cfraction } c \ 0 \geq 0$
 shows $h \ n \geq \text{fib } n$ <proof>

lemma *conv-denom-lower-bound*: $k \ n \geq \text{fib } (\text{Suc } n)$
<proof>

lemma *conv-diff-eq*: $\text{conv } c \ (\text{Suc } n) - \text{conv } c \ n = (-1) \wedge n / (k \ n * k \ (\text{Suc } n))$
<proof>

lemma *conv-telescope*:
 assumes $m \leq n$
 shows $\text{conv } c \ m + (\sum_{i=m..<n.} (-1) \wedge i / (k \ i * k \ (\text{Suc } i))) = \text{conv } c \ n$
<proof>

lemma *fib-at-top*: $\text{filterlim } \text{fib } \text{at-top } \text{at-top}$
<proof>

lemma *conv-denom-at-top*: $\text{filterlim } k \ \text{at-top } \text{at-top}$
<proof>

lemma

shows *summable-conv-telescope*:
 $\text{summable } (\lambda i. (-1) ^ i / (k i * k (Suc i)))$ (**is** *?th1*)

and *cfrac-remainder-bounds*:
 $|\sum i. (-1) ^ (i + m) / (k (i + m) * k (Suc i + m))| \in$
 $\{1 / (k m * (k m + k (Suc m))) <..< 1 / (k m * k (Suc m))\}$ (**is** *?th2*)

<proof>

lemma *convergent-conv*: *convergent* (*conv c*)
<proof>

lemma *LIMSEQ-cfrac-lim*: *cfrac-length c = ∞ \implies conv c \longrightarrow cfrac-lim c
*<proof>**

lemma *cfrac-lim-nonneg*:
assumes *cfrac-nth c 0 \geq 0*
shows *cfrac-lim c \geq 0*
<proof>

lemma *sums-cfrac-lim-minus-conv*:
assumes *cfrac-length c = ∞*
shows $(\lambda i. (-1) ^ (i + m) / (k (i + m) * k (Suc i + m))) \text{ sums } (cfrac-lim c -$
conv c m)
<proof>

lemma *cfrac-lim-minus-conv-upper-bound*:
assumes *m \leq cfrac-length c*
shows $|cfrac-lim c - conv c m| \leq 1 / (k m * k (Suc m))$
<proof>

lemma *cfrac-lim-minus-conv-lower-bound*:
assumes *m < cfrac-length c*
shows $|cfrac-lim c - conv c m| \geq 1 / (k m * (k m + k (Suc m)))$
<proof>

lemma *cfrac-lim-minus-conv-bounds*:
assumes *m < cfrac-length c*
shows $|cfrac-lim c - conv c m| \in \{1 / (k m * (k m + k (Suc m)))..1 / (k m$
 $* k (Suc m))\}$
<proof>

end

lemma *conv-pos'*:
assumes *n > 0 cfrac-nth c 0 \geq 0*
shows *conv c n > 0*
<proof>

lemma *conv-in-Rats* [*intro*]: *conv c n \in \mathbb{Q}*

<proof>

lemma

assumes $0 < z1 \ z1 \leq z2$

shows *conv'-even-mono*: $even \ n \implies conv' \ c \ n \ z1 \leq conv' \ c \ n \ z2$

and *conv'-odd-mono*: $odd \ n \implies conv' \ c \ n \ z1 \geq conv' \ c \ n \ z2$

<proof>

lemma

shows *conv-even-mono*: $even \ n \implies n \leq m \implies conv \ c \ n \leq conv \ c \ m$

and *conv-odd-mono*: $odd \ n \implies n \leq m \implies conv \ c \ n \geq conv \ c \ m$

<proof>

lemma

assumes $m \leq cfrac\text{-length} \ c$

shows *conv-le-cfrac-lim*: $even \ m \implies conv \ c \ m \leq cfrac\text{-lim} \ c$

and *conv-ge-cfrac-lim*: $odd \ m \implies conv \ c \ m \geq cfrac\text{-lim} \ c$

<proof>

lemma *cfrac-lim-ge-first*: $cfrac\text{-lim} \ c \geq cfrac\text{-nth} \ c \ 0$

<proof>

lemma *cfrac-lim-pos*: $cfrac\text{-nth} \ c \ 0 > 0 \implies cfrac\text{-lim} \ c > 0$

<proof>

lemma *conv'-eq-iff*:

assumes $0 \leq z1 \ \vee \ 0 \leq z2$

shows $conv' \ c \ n \ z1 = conv' \ c \ n \ z2 \longleftrightarrow z1 = z2$

<proof>

lemma *conv-even-mono-strict*:

assumes $even \ n \ n < m$

shows $conv \ c \ n < conv \ c \ m$

<proof>

lemma *conv-odd-mono-strict*:

assumes $odd \ n \ n < m$

shows $conv \ c \ n > conv \ c \ m$

<proof>

lemma *conv-less-cfrac-lim*:

assumes $even \ n \ n < cfrac\text{-length} \ c$

shows $conv \ c \ n < cfrac\text{-lim} \ c$

<proof>

lemma *conv-gt-cfrac-lim*:

assumes $odd \ n \ n < cfrac\text{-length} \ c$

shows $conv \ c \ n > cfrac\text{-lim} \ c$

<proof>

lemma *conv-neq-cfrac-lim*:
assumes $n < \text{cfrac-length } c$
shows $\text{conv } c \ n \neq \text{cfrac-lim } c$
 $\langle \text{proof} \rangle$

lemma *conv-ge-first*: $\text{conv } c \ n \geq \text{cfrac-nth } c \ 0$
 $\langle \text{proof} \rangle$

definition *cfrac-is-zero* :: $\text{cfrac} \Rightarrow \text{bool}$ **where** $\text{cfrac-is-zero } c \longleftrightarrow c = 0$

lemma *cfrac-is-zero-code* [code]: $\text{cfrac-is-zero } (\text{CFrac } n \ xs) \longleftrightarrow \text{lnull } xs \wedge n = 0$
 $\langle \text{proof} \rangle$

definition *cfrac-is-int* **where** $\text{cfrac-is-int } c \longleftrightarrow \text{cfrac-length } c = 0$

lemma *cfrac-is-int-code* [code]: $\text{cfrac-is-int } (\text{CFrac } n \ xs) \longleftrightarrow \text{lnull } xs$
 $\langle \text{proof} \rangle$

lemma *cfrac-length-of-int* [simp]: $\text{cfrac-length } (\text{cfrac-of-int } n) = 0$
 $\langle \text{proof} \rangle$

lemma *cfrac-is-int-of-int* [simp, intro]: $\text{cfrac-is-int } (\text{cfrac-of-int } n)$
 $\langle \text{proof} \rangle$

lemma *cfrac-is-int-iff*: $\text{cfrac-is-int } c \longleftrightarrow (\exists n. c = \text{cfrac-of-int } n)$
 $\langle \text{proof} \rangle$

lemma *cfrac-lim-reduce*:
assumes $\neg \text{cfrac-is-int } c$
shows $\text{cfrac-lim } c = \text{cfrac-nth } c \ 0 + 1 / \text{cfrac-lim } (\text{cfrac-tl } c)$
 $\langle \text{proof} \rangle$

lemma *cfrac-lim-tl*:
assumes $\neg \text{cfrac-is-int } c$
shows $\text{cfrac-lim } (\text{cfrac-tl } c) = 1 / (\text{cfrac-lim } c - \text{cfrac-nth } c \ 0)$
 $\langle \text{proof} \rangle$

lemma *cfrac-remainder-Suc'*:
assumes $n < \text{cfrac-length } c$
shows $\text{cfrac-remainder } c \ (\text{Suc } n) * (\text{cfrac-remainder } c \ n - \text{cfrac-nth } c \ n) = 1$
 $\langle \text{proof} \rangle$

lemma *cfrac-remainder-Suc*:
assumes $n < \text{cfrac-length } c$

shows $\text{cfrac-remainder } c \text{ (Suc } n) = 1 / (\text{cfrac-remainder } c \text{ } n - \text{cfrac-nth } c \text{ } n)$
 ⟨proof⟩

lemma *cfrac-remainder-0* [simp]: $\text{cfrac-remainder } c \text{ } 0 = \text{cfrac-lim } c$
 ⟨proof⟩

context

fixes $c \ h \ k \ x$

defines $h \equiv \text{conv-num } c$ **and** $k \equiv \text{conv-denom } c$ **and** $x \equiv \text{cfrac-remainder } c$

begin

lemma *cfrac-lim-eq-num-denom-remainder-aux*:

assumes $\text{Suc } (\text{Suc } n) \leq \text{cfrac-length } c$

shows $\text{cfrac-lim } c * (k \text{ (Suc } n) * x \text{ (Suc (Suc } n)) + k \text{ } n) = h \text{ (Suc } n) * x \text{ (Suc (Suc } n)) + h \text{ } n$

⟨proof⟩

lemma *cfrac-remainder-nonneg*: $\text{cfrac-nth } c \text{ } n \geq 0 \implies \text{cfrac-remainder } c \text{ } n \geq 0$
 ⟨proof⟩

lemma *cfrac-remainder-pos*: $\text{cfrac-nth } c \text{ } n > 0 \implies \text{cfrac-remainder } c \text{ } n > 0$
 ⟨proof⟩

lemma *cfrac-lim-eq-num-denom-remainder*:

assumes $\text{Suc } (\text{Suc } n) < \text{cfrac-length } c$

shows $\text{cfrac-lim } c = (h \text{ (Suc } n) * x \text{ (Suc (Suc } n)) + h \text{ } n) / (k \text{ (Suc } n) * x \text{ (Suc (Suc } n)) + k \text{ } n)$

⟨proof⟩

lemma *abs-diff-successive-conv*:

shows $|\text{conv } c \text{ (Suc } n) - \text{conv } c \text{ } n| = 1 / (k \text{ } n * k \text{ (Suc } n))$

⟨proof⟩

lemma *conv-denom-plus2-ratio-ge*: $k \text{ (Suc (Suc } n)) \geq 2 * k \text{ } n$
 ⟨proof⟩

end

lemma *conv'-cfrac-remainder*:

assumes $n < \text{cfrac-length } c$

shows $\text{conv}' c \text{ } n \text{ (cfrac-remainder } c \text{ } n) = \text{cfrac-lim } c$

⟨proof⟩

lemma *cfrac-lim-rational* [intro]:

assumes $\text{cfrac-length } c < \infty$

shows $\text{cfrac-lim } c \in \mathbf{Q}$

⟨proof⟩

lemma *linfinite-cfrac-of-real-aux*:

$x \notin \mathbb{Q} \implies x \in \{0 < \dots < 1\} \implies \text{linfinite } (\text{cfrac-of-real-aux } x)$
 ⟨proof⟩

lemma *cfrac-length-of-real-irrational*:
 assumes $x \notin \mathbb{Q}$
 shows $\text{cfrac-length } (\text{cfrac-of-real } x) = \infty$
 ⟨proof⟩

lemma *cfrac-length-of-real-reduce*:
 assumes $x \notin \mathbb{Z}$
 shows $\text{cfrac-length } (\text{cfrac-of-real } x) = \text{eSuc } (\text{cfrac-length } (\text{cfrac-of-real } (1 / \text{frac } x)))$
 ⟨proof⟩

lemma *cfrac-length-of-real-int [simp]*: $x \in \mathbb{Z} \implies \text{cfrac-length } (\text{cfrac-of-real } x) = 0$
 ⟨proof⟩

lemma *conv-cfrac-of-real-le-ge*:
 assumes $n \leq \text{cfrac-length } (\text{cfrac-of-real } x)$
 shows *if even n then $\text{conv } (\text{cfrac-of-real } x) n \leq x$ else $\text{conv } (\text{cfrac-of-real } x) n \geq x$*
 ⟨proof⟩

lemma *cfrac-lim-of-real [simp]*: $\text{cfrac-lim } (\text{cfrac-of-real } x) = x$
 ⟨proof⟩

lemma *Ints-add-left-cancel*: $x \in \mathbb{Z} \implies x + y \in \mathbb{Z} \longleftrightarrow y \in \mathbb{Z}$
 ⟨proof⟩

lemma *Ints-add-right-cancel*: $y \in \mathbb{Z} \implies x + y \in \mathbb{Z} \longleftrightarrow x \in \mathbb{Z}$
 ⟨proof⟩

lemma *cfrac-of-real-conv'*:
 fixes $m n :: \text{nat}$
 assumes $x > 1$ $m < n$
 shows $\text{cfrac-nth } (\text{cfrac-of-real } (\text{conv}' c n x)) m = \text{cfrac-nth } c m$
 ⟨proof⟩

lemma *cfrac-lim-irrational*:
 assumes [simp]: $\text{cfrac-length } c = \infty$
 shows $\text{cfrac-lim } c \notin \mathbb{Q}$
 ⟨proof⟩

lemma *cfrac-infinite-iff*: $\text{cfrac-length } c = \infty \longleftrightarrow \text{cfrac-lim } c \notin \mathbb{Q}$
 ⟨proof⟩

lemma *cfrac-lim-rational-iff*: $\text{cfrac-lim } c \in \mathbb{Q} \longleftrightarrow \text{cfrac-length } c \neq \infty$
 ⟨proof⟩

lemma *cfrac-of-real-infinite-iff* [simp]: $cfrac\text{-length } (cfrac\text{-of-real } x) = \infty \longleftrightarrow x \notin \mathbf{Q}$

<proof>

lemma *cfrac-remainder-rational-iff* [simp]:
 $cfrac\text{-remainder } c \ n \in \mathbf{Q} \longleftrightarrow cfrac\text{-length } c < \infty$

<proof>

lift-definition *cfrac-cons* :: $int \Rightarrow cfrac \Rightarrow cfrac$ **is**

$\lambda a \ bs.$ case *bs* of $(b, bs) \Rightarrow$ if $b \leq 0$ then $(1, LNil)$ else $(a, LCons (nat (b - 1)) bs)$ *<proof>*

lemma *cfrac-nth-cons*:

assumes $cfrac\text{-nth } x \ 0 \geq 1$

shows $cfrac\text{-nth } (cfrac\text{-cons } a \ x) \ n = (if \ n = 0 \ then \ a \ else \ cfrac\text{-nth } x \ (n - 1))$

<proof>

lemma *cfrac-length-cons* [simp]:

assumes $cfrac\text{-nth } x \ 0 \geq 1$

shows $cfrac\text{-length } (cfrac\text{-cons } a \ x) = eSuc (cfrac\text{-length } x)$

<proof>

lemma *cfrac-tl-cons* [simp]:

assumes $cfrac\text{-nth } x \ 0 \geq 1$

shows $cfrac\text{-tl } (cfrac\text{-cons } a \ x) = x$

<proof>

lemma *cfrac-cons-tl*:

assumes $\neg cfrac\text{-is-int } x$

shows $cfrac\text{-cons } (cfrac\text{-nth } x \ 0) (cfrac\text{-tl } x) = x$

<proof>

1.3 Non-canonical continued fractions

As we will show later, every irrational number has a unique continued fraction expansion. Every rational number x , however, has two different expansions: The canonical one ends with some number n (which is not equal to 1 unless $x = 1$) and a non-canonical one which ends with $n - 1, 1$.

We now define this non-canonical expansion analogously to the canonical one before and show its characteristic properties:

- The length of the non-canonical expansion is one greater than that of the canonical one.
- If the expansion is infinite, the non-canonical and the canonical one coincide.
- The coefficients of the expansions are all equal except for the last two.

The last coefficient of the non-canonical expansion is always 1, and the second to last one is the last of the canonical one minus 1.

lift-definition *cfrac-canonical* :: *cfrac* \Rightarrow *bool* **is**

$\lambda(x, xs). \neg \text{lfinite } xs \vee \text{lnull } xs \vee \text{llast } xs \neq 0$ \langle proof \rangle

lemma *cfrac-canonical* [code]:

cfrac-canonical (*CFrac* *x xs*) \longleftrightarrow *lnull* *xs* \vee *llast* *xs* $\neq 0 \vee \neg \text{lfinite } xs$
 \langle proof \rangle

lemma *cfrac-canonical-iff*:

cfrac-canonical *c* \longleftrightarrow
cfrac-length *c* $\in \{0, \infty\} \vee \text{cfrac-nth } c$ (*the-enat* (*cfrac-length* *c*)) $\neq 1$
 \langle proof \rangle

lemma *llast-cfrac-of-real-aux-nonzero*:

assumes *lfinite* (*cfrac-of-real-aux* *x*) $\neg \text{lnull}$ (*cfrac-of-real-aux* *x*)
shows *llast* (*cfrac-of-real-aux* *x*) $\neq 0$
 \langle proof \rangle

lemma *cfrac-canonical-of-real* [intro]: *cfrac-canonical* (*cfrac-of-real* *x*)

\langle proof \rangle

primcorec *cfrac-of-real-alt-aux* :: *real* \Rightarrow *nat* *llist* **where**

cfrac-of-real-alt-aux *x* =
 (if *x* $\in \{0 < .. < 1\}$ then
 if $1 / x \in \mathbb{Z}$ then
 LCons (*nat* $\lfloor 1/x \rfloor - 2$) (*LCons* 0 *LNil*)
 else *LCons* (*nat* $\lfloor 1/x \rfloor - 1$) (*cfrac-of-real-alt-aux* (*frac* ($1/x$)))
 else *LNil*)

lemma *cfrac-of-real-aux-alt-LNil* [simp]: $x \notin \{0 < .. < 1\} \implies \text{cfrac-of-real-alt-aux } x = \text{LNil}$

\langle proof \rangle

lemma *cfrac-of-real-aux-alt-0* [simp]: *cfrac-of-real-alt-aux* 0 = *LNil*

\langle proof \rangle

lemma *cfrac-of-real-aux-alt-eq-LNil-iff* [simp]: *cfrac-of-real-alt-aux* *x* = *LNil* \longleftrightarrow $x \notin \{0 < .. < 1\}$

\langle proof \rangle

lift-definition *cfrac-of-real-alt* :: *real* \Rightarrow *cfrac* **is**

$\lambda x. \text{if } x \in \mathbb{Z} \text{ then } (\lfloor x \rfloor - 1, \text{LCons } 0 \text{ LNil}) \text{ else } (\lfloor x \rfloor, \text{cfrac-of-real-alt-aux } (\text{frac } x))$ \langle proof \rangle

lemma *cfrac-tl-of-real-alt*:

assumes $x \notin \mathbb{Z}$

shows *cfrac-tl* (*cfrac-of-real-alt* *x*) = *cfrac-of-real-alt* ($1 / \text{frac } x$)

<proof>

lemma *cfrac-nth-of-real-alt-Suc*:

assumes $x \notin \mathbb{Z}$

shows $\text{cfrac-nth } (\text{cfrac-of-real-alt } x) (\text{Suc } n) = \text{cfrac-nth } (\text{cfrac-of-real-alt } (1 / \text{frac } x)) n$

<proof>

lemma *cfrac-nth-gt0-of-real-int [simp]*:

$m > 0 \implies \text{cfrac-nth } (\text{cfrac-of-real } (\text{of-int } n)) m = 1$

<proof>

lemma *cfrac-nth-0-of-real-alt-int [simp]*:

$\text{cfrac-nth } (\text{cfrac-of-real-alt } (\text{of-int } n)) 0 = n - 1$

<proof>

lemma *cfrac-nth-gt0-of-real-alt-int [simp]*:

$m > 0 \implies \text{cfrac-nth } (\text{cfrac-of-real-alt } (\text{of-int } n)) m = 1$

<proof>

lemma *llength-cfrac-of-real-alt-aux*:

assumes $x \in \{0 < .. < 1\}$

shows $\text{llength } (\text{cfrac-of-real-alt-aux } x) = \text{eSuc } (\text{llength } (\text{cfrac-of-real-aux } x))$

<proof>

lemma *cfrac-length-of-real-alt*:

$\text{cfrac-length } (\text{cfrac-of-real-alt } x) = \text{eSuc } (\text{cfrac-length } (\text{cfrac-of-real } x))$

<proof>

lemma *cfrac-of-real-alt-aux-eq-regular*:

assumes $x \in \{0 < .. < 1\}$ $\text{llength } (\text{cfrac-of-real-aux } x) = \infty$

shows $\text{cfrac-of-real-alt-aux } x = \text{cfrac-of-real-aux } x$

<proof>

lemma *cfrac-of-real-alt-irrational [simp]*:

assumes $x \notin \mathbb{Q}$

shows $\text{cfrac-of-real-alt } x = \text{cfrac-of-real } x$

<proof>

lemma *cfrac-nth-of-real-alt-0*:

$\text{cfrac-nth } (\text{cfrac-of-real-alt } x) 0 = (\text{if } x \in \mathbb{Z} \text{ then } \lfloor x \rfloor - 1 \text{ else } \lfloor x \rfloor)$

<proof>

lemma *cfrac-nth-of-real-alt*:

fixes $n :: \text{nat}$ **and** $x :: \text{real}$

defines $c \equiv \text{cfrac-of-real } x$

defines $c' \equiv \text{cfrac-of-real-alt } x$

defines $l \equiv \text{cfrac-length } c$

shows $\text{cfrac-nth } c' n =$

(if enat $n = l$ then
 cfrac-nth $c\ n - 1$
 else if enat $n = l + 1$ then
 1
 else
 cfrac-nth $c\ n$)
 ⟨proof⟩

lemma *cfrac-of-real-length-eq-0-iff*: cfrac-length (cfrac-of-real x) = 0 \longleftrightarrow $x \in \mathbb{Z}$
 ⟨proof⟩

lemma *conv'-cong*:
 assumes $(\bigwedge k. k < n \implies \text{cfrac-nth } c\ k = \text{cfrac-nth } c'\ k)$ $n = n'$ $x = y$
 shows $\text{conv}'\ c\ n\ x = \text{conv}'\ c'\ n'\ y$
 ⟨proof⟩

lemma *conv-cong*:
 assumes $(\bigwedge k. k \leq n \implies \text{cfrac-nth } c\ k = \text{cfrac-nth } c'\ k)$ $n = n'$
 shows $\text{conv } c\ n = \text{conv } c'\ n'$
 ⟨proof⟩

lemma *conv'-cfrac-of-real-alt*:
 assumes $\text{enat } n \leq \text{cfrac-length } (\text{cfrac-of-real } x)$
 shows $\text{conv}'\ (\text{cfrac-of-real-alt } x)\ n\ y = \text{conv}'\ (\text{cfrac-of-real } x)\ n\ y$
 ⟨proof⟩

lemma *cfrac-lim-of-real-alt [simp]*: cfrac-lim (cfrac-of-real-alt x) = x
 ⟨proof⟩

lemma *cfrac-eqI*:
 assumes $\text{cfrac-length } c = \text{cfrac-length } c'$ and $\bigwedge n. \text{cfrac-nth } c\ n = \text{cfrac-nth } c'\ n$
 shows $c = c'$
 ⟨proof⟩

lemma *cfrac-eq-0I*:
 assumes $\text{cfrac-lim } c = 0$ $\text{cfrac-nth } c\ 0 \geq 0$
 shows $c = 0$
 ⟨proof⟩

lemma *cfrac-eq-1I*:
 assumes $\text{cfrac-lim } c = 1$ $\text{cfrac-nth } c\ 0 \neq 0$
 shows $c = 1$
 ⟨proof⟩

lemma *cfrac-coinduct [coinduct type: cfrac]*:
 assumes $R\ c1\ c2$
 assumes IH: $\bigwedge c1\ c2. R\ c1\ c2 \implies$
 $\text{cfrac-is-int } c1 = \text{cfrac-is-int } c2 \wedge$
 $\text{cfrac-nth } c1\ 0 = \text{cfrac-nth } c2\ 0 \wedge$

$(\neg \text{cfrac-is-int } c1 \longrightarrow \neg \text{cfrac-is-int } c2 \longrightarrow R (\text{cfrac-tl } c1) (\text{cfrac-tl } c2))$
shows $c1 = c2$
 <proof>

lemma *cfrac-nth-0-cases*:

$\text{cfrac-nth } c \ 0 = \lfloor \text{cfrac-lim } c \rfloor \vee \text{cfrac-nth } c \ 0 = \lfloor \text{cfrac-lim } c \rfloor - 1 \wedge \text{cfrac-tl } c = 1$
 <proof>

lemma *cfrac-length-1 [simp]*: $\text{cfrac-length } 1 = 0$
 <proof>

lemma *cfrac-nth-1 [simp]*: $\text{cfrac-nth } 1 \ m = 1$
 <proof>

lemma *cfrac-lim-1 [simp]*: $\text{cfrac-lim } 1 = 1$
 <proof>

lemma *cfrac-nth-0-not-int*:

assumes $\text{cfrac-lim } c \notin \mathbb{Z}$
shows $\text{cfrac-nth } c \ 0 = \lfloor \text{cfrac-lim } c \rfloor$
 <proof>

lemma *cfrac-of-real-cfrac-lim-irrational*:

assumes $\text{cfrac-lim } c \notin \mathbb{Q}$
shows $\text{cfrac-of-real } (\text{cfrac-lim } c) = c$
 <proof>

lemma *cfrac-eqI-first*:

assumes $\neg \text{cfrac-is-int } c \ \neg \text{cfrac-is-int } c'$
assumes $\text{cfrac-nth } c \ 0 = \text{cfrac-nth } c' \ 0$ **and** $\text{cfrac-tl } c = \text{cfrac-tl } c'$
shows $c = c'$
 <proof>

lemma *cfrac-is-int-of-real-iff*: $\text{cfrac-is-int } (\text{cfrac-of-real } x) \longleftrightarrow x \in \mathbb{Z}$
 <proof>

lemma *cfrac-not-is-int-of-real-alt*: $\neg \text{cfrac-is-int } (\text{cfrac-of-real-alt } x)$
 <proof>

lemma *cfrac-tl-of-real-alt-of-int [simp]*: $\text{cfrac-tl } (\text{cfrac-of-real-alt } (\text{of-int } n)) = 1$
 <proof>

lemma *cfrac-is-intI*:

assumes $\text{cfrac-nth } c \ 0 \geq \lfloor \text{cfrac-lim } c \rfloor$ **and** $\text{cfrac-lim } c \in \mathbb{Z}$
shows $\text{cfrac-is-int } c$
 <proof>

lemma *cfrac-eq-of-intI*:

assumes *cfrac-nth* c $0 \geq \lfloor \text{cfrac-lim } c \rfloor$ **and** *cfrac-lim* $c \in \mathbb{Z}$

shows $c = \text{cfrac-of-int } \lfloor \text{cfrac-lim } c \rfloor$

<proof>

lemma *cfrac-lim-of-int [simp]*: *cfrac-lim* (*cfrac-of-int* n) = *of-int* n

<proof>

lemma *cfrac-of-real-of-int [simp]*: *cfrac-of-real* (*of-int* n) = *cfrac-of-int* n

<proof>

lemma *cfrac-of-real-of-nat [simp]*: *cfrac-of-real* (*of-nat* n) = *cfrac-of-int* (*int* n)

<proof>

lemma *cfrac-int-cases*:

assumes *cfrac-lim* $c = \text{of-int } n$

shows $c = \text{cfrac-of-int } n \vee c = \text{cfrac-of-real-alt } (\text{of-int } n)$

<proof>

lemma *cfrac-cases*:

$c \in \{\text{cfrac-of-real } (\text{cfrac-lim } c), \text{cfrac-of-real-alt } (\text{cfrac-lim } c)\}$

<proof>

lemma *cfrac-lim-eq-iff*:

assumes *cfrac-length* $c = \infty \vee \text{cfrac-length } c' = \infty$

shows $\text{cfrac-lim } c = \text{cfrac-lim } c' \longleftrightarrow c = c'$

<proof>

lemma *floor-cfrac-remainder*:

assumes *cfrac-length* $c = \infty$

shows $\lfloor \text{cfrac-remainder } c \ n \rfloor = \text{cfrac-nth } c \ n$

<proof>

1.4 Approximation properties

In this section, we will show that convergents of the continued fraction expansion of a number x are good approximations of x , and in a certain sense, the reverse holds as well.

lemma *sgn-of-int*:

sgn (*of-int* $x :: 'a :: \{\text{linordered-idom}\}$) = *of-int* (*sgn* x)

<proof>

lemma *conv-ge-one*: *cfrac-nth* c $0 > 0 \implies \text{conv } c \ n \geq 1$

<proof>

context

fixes $c \ h \ k$

defines $h \equiv \text{conv-num } c$ **and** $k \equiv \text{conv-denom } c$

begin

lemma *abs-diff-le-abs-add*:
fixes $x\ y :: \text{real}$
assumes $x \geq 0 \wedge y \geq 0 \vee x \leq 0 \wedge y \leq 0$
shows $|x - y| \leq |x + y|$
 $\langle \text{proof} \rangle$

lemma *abs-diff-less-abs-add*:
fixes $x\ y :: \text{real}$
assumes $x > 0 \wedge y > 0 \vee x < 0 \wedge y < 0$
shows $|x - y| < |x + y|$
 $\langle \text{proof} \rangle$

lemma *abs-diff-le-imp-same-sign*:
assumes $|x - y| \leq d\ d < |y|$
shows $\text{sgn } x = \text{sgn } (y::\text{real})$
 $\langle \text{proof} \rangle$

lemma *conv-nonpos*:
assumes $\text{cfrac-nth } c\ 0 < 0$
shows $\text{conv } c\ n \leq 0$
 $\langle \text{proof} \rangle$

lemma *cfrac-lim-nonpos*:
assumes $\text{cfrac-nth } c\ 0 < 0$
shows $\text{cfrac-lim } c \leq 0$
 $\langle \text{proof} \rangle$

lemma *conv-num-nonpos*:
assumes $\text{cfrac-nth } c\ 0 < 0$
shows $h\ n \leq 0$
 $\langle \text{proof} \rangle$

lemma *conv-best-approximation-aux*:
 $\text{cfrac-lim } c \geq 0 \wedge h\ n \geq 0 \vee \text{cfrac-lim } c \leq 0 \wedge h\ n \leq 0$
 $\langle \text{proof} \rangle$

lemma *conv-best-approximation-ex*:
fixes $a\ b :: \text{int}$ **and** $x :: \text{real}$
assumes $n \leq \text{cfrac-length } c$
assumes $0 < b$ **and** $b \leq k\ n$ **and** *coprime* $a\ b$ **and** $n > 0$
assumes $(a, b) \neq (h\ n, k\ n)$
assumes $\neg(\text{cfrac-length } c = 1 \wedge n = 0)$
assumes $\text{Suc } n \neq \text{cfrac-length } c \vee \text{cfrac-canonical } c$
defines $x \equiv \text{cfrac-lim } c$
shows $|k\ n * x - h\ n| < |b * x - a|$
 $\langle \text{proof} \rangle$

lemma *conv-best-approximation-ex-weak*:

fixes $a\ b :: \text{int}$ **and** $x :: \text{real}$
assumes $n \leq \text{cfrac-length } c$
assumes $0 < b$ **and** $b < k$ ($\text{Suc } n$) **and** $\text{coprime } a\ b$
defines $x \equiv \text{cfrac-lim } c$
shows $|k\ n * x - h\ n| \leq |b * x - a|$
<proof>

lemma *cfrac-canonical-reduce*:
 $\text{cfrac-canonical } c \longleftrightarrow$
 $\text{cfrac-is-int } c \vee \neg \text{cfrac-is-int } c \wedge \text{cfrac-tl } c \neq 1 \wedge \text{cfrac-canonical } (\text{cfrac-tl } c)$
<proof>

lemma *cfrac-nth-0-conv-floor*:
assumes $\text{cfrac-canonical } c \vee \text{cfrac-length } c \neq 1$
shows $\text{cfrac-nth } c\ 0 = \lfloor \text{cfrac-lim } c \rfloor$
<proof>

lemma *conv-best-approximation-ex-nat*:
fixes $a\ b :: \text{nat}$ **and** $x :: \text{real}$
assumes $n \leq \text{cfrac-length } c$ $0 < b$ $b < k$ ($\text{Suc } n$) $\text{coprime } a\ b$
shows $|k\ n * \text{cfrac-lim } c - h\ n| \leq |b * \text{cfrac-lim } c - a|$
<proof>

lemma *abs-mult-nonneg-left*:
assumes $x \geq (0 :: 'a :: \{\text{ordered-ab-group-add-abs, idom-abs-sgn}\})$
shows $x * |y| = |x * y|$
<proof>

Any convergent of the continued fraction expansion of x is a best approximation of x , i.e. there is no other number with a smaller denominator that approximates it better.

lemma *conv-best-approximation*:
fixes $a\ b :: \text{int}$ **and** $x :: \text{real}$
assumes $n \leq \text{cfrac-length } c$
assumes $0 < b$ **and** $b < k\ n$ **and** $\text{coprime } a\ b$
defines $x \equiv \text{cfrac-lim } c$
shows $|x - \text{conv } c\ n| \leq |x - a / b|$
<proof>

lemma *conv-denom-partition*:
assumes $y > 0$
shows $\exists! n. y \in \{k\ n..<k\ (\text{Suc } n)\}$
<proof>

A fraction that approximates a real number x sufficiently well (in a certain sense) is a convergent of its continued fraction expansion.

lemma *frac-is-convergentI*:
fixes $a\ b :: \text{int}$ **and** $x :: \text{real}$
defines $x \equiv \text{cfrac-lim } c$

assumes $b > 0$ **and** *coprime* a b **and** $|x - a / b| < 1 / (2 * b^2)$
shows $\exists n. \text{enat } n \leq \text{cfrac-length } c \wedge (a, b) = (h \ n, k \ n)$
 <proof>

end

1.5 Efficient code for convergents

function *conv-gen* :: $(\text{nat} \Rightarrow \text{int}) \Rightarrow \text{int} \times \text{int} \times \text{nat} \Rightarrow \text{nat} \Rightarrow \text{int}$ **where**
conv-gen c (a, b, n) $N =$
 (if $n > N$ then b else *conv-gen* c $(b, b * c \ n + a, \text{Suc } n)$ N)
 <proof>
termination <proof>

lemmas [*simp del*] = *conv-gen.simps*

lemma *conv-gen-aux-simps* [*simp*]:
 $n > N \Longrightarrow \text{conv-gen } c \ (a, b, n) \ N = b$
 $n \leq N \Longrightarrow \text{conv-gen } c \ (a, b, n) \ N = \text{conv-gen } c \ (b, b * c \ n + a, \text{Suc } n) \ N$
 <proof>

lemma *conv-num-eq-conv-gen-aux*:
 $\text{Suc } n \leq N \Longrightarrow \text{conv-num } c \ n = b * \text{cfrac-nth } c \ n + a \Longrightarrow$
 $\text{conv-num } c \ (\text{Suc } n) = \text{conv-num } c \ n * \text{cfrac-nth } c \ (\text{Suc } n) + b \Longrightarrow$
 $\text{conv-num } c \ N = \text{conv-gen } (\text{cfrac-nth } c) \ (a, b, n) \ N$
 <proof>

lemma *conv-denom-eq-conv-gen-aux*:
 $\text{Suc } n \leq N \Longrightarrow \text{conv-denom } c \ n = b * \text{cfrac-nth } c \ n + a \Longrightarrow$
 $\text{conv-denom } c \ (\text{Suc } n) = \text{conv-denom } c \ n * \text{cfrac-nth } c \ (\text{Suc } n) + b \Longrightarrow$
 $\text{conv-denom } c \ N = \text{conv-gen } (\text{cfrac-nth } c) \ (a, b, n) \ N$
 <proof>

lemma *conv-num-code* [*code*]: $\text{conv-num } c \ n = \text{conv-gen } (\text{cfrac-nth } c) \ (0, 1, 0) \ n$
 <proof>

lemma *conv-denom-code* [*code*]: $\text{conv-denom } c \ n = \text{conv-gen } (\text{cfrac-nth } c) \ (1, 0, 0) \ n$
 <proof>

definition *conv-num-fun* **where** $\text{conv-num-fun } c = \text{conv-gen } c \ (0, 1, 0)$

definition *conv-denom-fun* **where** $\text{conv-denom-fun } c = \text{conv-gen } c \ (1, 0, 0)$

lemma

assumes *is-cfrac* c

shows *conv-num-fun-eq*: $\text{conv-num-fun } c \ n = \text{conv-num } (\text{cfrac } c) \ n$

and *conv-denom-fun-eq*: $\text{conv-denom-fun } c \ n = \text{conv-denom } (\text{cfrac } c) \ n$

<proof>

1.6 Computing the continued fraction expansion of a rational number

function *cfrac-list-of-rat* :: *int* × *int* ⇒ *int list* **where**

cfrac-list-of-rat (*a*, *b*) =
 (if *b* = 0 then [0]
 else *a div b* # (if *a mod b* = 0 then [] else *cfrac-list-of-rat* (*b*, *a mod b*)))
 ⟨*proof*⟩

termination

⟨*proof*⟩

lemmas [*simp del*] = *cfrac-list-of-rat.simps*

lemma *cfrac-list-of-rat-correct*:

(let *xs* = *cfrac-list-of-rat* (*a*, *b*); *c* = *cfrac-of-real* (*a* / *b*)
 in *length xs* = *cfrac-length c* + 1 ∧ (∀ *i* < *length xs*. *xs* ! *i* = *cfrac-nth c i*)
 ⟨*proof*⟩

lemma *conv-num-cong*:

assumes (∧ *k*. *k* ≤ *n* ⇒ *cfrac-nth c k* = *cfrac-nth c' k*) *n* = *n'*
shows *conv-num c n* = *conv-num c' n*
 ⟨*proof*⟩

lemma *conv-denom-cong*:

assumes (∧ *k*. *k* ≤ *n* ⇒ *cfrac-nth c k* = *cfrac-nth c' k*) *n* = *n'*
shows *conv-denom c n* = *conv-denom c' n'*
 ⟨*proof*⟩

lemma *cfrac-lim-diff-le*:

assumes ∀ *k* ≤ *Suc n*. *cfrac-nth c1 k* = *cfrac-nth c2 k*
assumes *n* ≤ *cfrac-length c1* *n* ≤ *cfrac-length c2*
shows |*cfrac-lim c1* - *cfrac-lim c2*| ≤ 2 / (*conv-denom c1 n* * *conv-denom c1*
 (*Suc n*))
 ⟨*proof*⟩

lemma *of-int-leI*: *n* ≤ *m* ⇒ (*of-int n* :: '*a* :: *linordered-idom*) ≤ *of-int m*

⟨*proof*⟩

lemma *cfrac-lim-diff-le'*:

assumes ∀ *k* ≤ *Suc n*. *cfrac-nth c1 k* = *cfrac-nth c2 k*
assumes *n* ≤ *cfrac-length c1* *n* ≤ *cfrac-length c2*
shows |*cfrac-lim c1* - *cfrac-lim c2*| ≤ 2 / (*fib (n+1)* * *fib (n+2)*)
 ⟨*proof*⟩

end

2 Quadratic Irrationals

theory *Quadratic-Irrationals*

```

imports
  Continued-Fractions
  HOL-Computational-Algebra.Computational-Algebra
  HOL-Library.Discrete-Functions
  Coinductive.Coinductive-Stream
begin

```

```

lemma snth-cycle:
  assumes  $xs \neq []$ 
  shows  $snth (cycle xs) n = xs ! (n \bmod length xs)$ 
<proof>

```

2.1 Basic results on rationality of square roots

```

lemma inverse-in-Rats-iff [simp]:  $inverse (x :: real) \in \mathbb{Q} \longleftrightarrow x \in \mathbb{Q}$ 
<proof>

```

```

lemma nonneg-sqrt-nat-or-irrat:
  assumes  $x^2 = real a$  and  $x \geq 0$ 
  shows  $x \in \mathbb{N} \vee x \notin \mathbb{Q}$ 
<proof>

```

A square root of a natural number is either an integer or irrational.

```

corollary sqrt-nat-or-irrat:
  assumes  $x^2 = real a$ 
  shows  $x \in \mathbb{Z} \vee x \notin \mathbb{Q}$ 
<proof>

```

```

corollary sqrt-nat-or-irrat':
   $sqrt (real a) \in \mathbb{N} \vee sqrt (real a) \notin \mathbb{Q}$ 
<proof>

```

The square root of a natural number n is again a natural number iff n is a perfect square.

```

corollary sqrt-nat-iff-is-square:
   $sqrt (real n) \in \mathbb{N} \longleftrightarrow is-square n$ 
<proof>

```

```

corollary irrat-sqrt-nonsquare:  $\neg is-square n \implies sqrt (real n) \notin \mathbb{Q}$ 
<proof>

```

```

lemma sqrt-of-nat-in-Rats-iff:  $sqrt (real n) \in \mathbb{Q} \longleftrightarrow is-square n$ 
<proof>

```

```

lemma Discrete-sqrt-altdef:  $floor-sqrt n = nat \lfloor sqrt n \rfloor$ 
<proof>

```


2.2 Definition of quadratic irrationals

Irrational real numbers x that satisfy a quadratic equation $ax^2 + bx + c = 0$ with a, b, c not all equal to 0 are called *quadratic irrationals*. These are of the form $p + q\sqrt{d}$ for rational numbers p, q and a positive integer d .

inductive *quadratic-irrational* :: real \Rightarrow bool **where**

$x \notin \mathbf{Q} \implies \text{real-of-int } a * x^2 + \text{real-of-int } b * x + \text{real-of-int } c = 0 \implies$
 $a \neq 0 \vee b \neq 0 \vee c \neq 0 \implies \text{quadratic-irrational } x$

lemma *quadratic-irrational-sqrt* [intro]:

assumes $\neg \text{is-square } n$

shows *quadratic-irrational* (sqrt (real n))

$\langle \text{proof} \rangle$

lemma *quadratic-irrational-uminus* [intro]:

assumes *quadratic-irrational* x

shows *quadratic-irrational* ($-x$)

$\langle \text{proof} \rangle$

lemma *quadratic-irrational-uminus-iff* [simp]:

quadratic-irrational ($-x$) \longleftrightarrow *quadratic-irrational* x

$\langle \text{proof} \rangle$

lemma *quadratic-irrational-plus-int* [intro]:

assumes *quadratic-irrational* x

shows *quadratic-irrational* ($x + \text{of-int } n$)

$\langle \text{proof} \rangle$

lemma *quadratic-irrational-plus-int-iff* [simp]:

quadratic-irrational ($x + \text{of-int } n$) \longleftrightarrow *quadratic-irrational* x

$\langle \text{proof} \rangle$

lemma *quadratic-irrational-minus-int-iff* [simp]:

quadratic-irrational ($x - \text{of-int } n$) \longleftrightarrow *quadratic-irrational* x

$\langle \text{proof} \rangle$

lemma *quadratic-irrational-plus-nat-iff* [simp]:

quadratic-irrational ($x + \text{of-nat } n$) \longleftrightarrow *quadratic-irrational* x

$\langle \text{proof} \rangle$

lemma *quadratic-irrational-minus-nat-iff* [simp]:

quadratic-irrational ($x - \text{of-nat } n$) \longleftrightarrow *quadratic-irrational* x

$\langle \text{proof} \rangle$

lemma *quadratic-irrational-plus-1-iff* [simp]:

quadratic-irrational ($x + 1$) \longleftrightarrow *quadratic-irrational* x

$\langle \text{proof} \rangle$

lemma *quadratic-irrational-minus-1-iff* [simp]:

quadratic-irrational $(x - 1) \longleftrightarrow$ *quadratic-irrational* x
 ⟨proof⟩

lemma *quadratic-irrational-plus-numeral-iff* [simp]:
quadratic-irrational $(x + \text{numeral } n) \longleftrightarrow$ *quadratic-irrational* x
 ⟨proof⟩

lemma *quadratic-irrational-minus-numeral-iff* [simp]:
quadratic-irrational $(x - \text{numeral } n) \longleftrightarrow$ *quadratic-irrational* x
 ⟨proof⟩

lemma *quadratic-irrational-inverse*:
assumes *quadratic-irrational* x
shows *quadratic-irrational* $(\text{inverse } x)$
 ⟨proof⟩

lemma *quadratic-irrational-inverse-iff* [simp]:
quadratic-irrational $(\text{inverse } x) \longleftrightarrow$ *quadratic-irrational* x
 ⟨proof⟩

lemma *quadratic-irrational-cfrac-remainder-iff*:
quadratic-irrational $(\text{cfrac-remainder } c \ n) \longleftrightarrow$ *quadratic-irrational* $(\text{cfrac-lim } c)$
 ⟨proof⟩

2.3 Real solutions of quadratic equations

For the next result, we need some basic properties of real solutions to quadratic equations.

lemma *quadratic-equation-reals*:
fixes $a \ b \ c :: \text{real}$
defines $f \equiv (\lambda x. a * x^2 + b * x + c)$
defines $\text{discr} \equiv (b^2 - 4 * a * c)$
shows $\{x. f \ x = 0\} =$
 (if $a = 0$ then
 (if $b = 0$ then if $c = 0$ then UNIV else {} else $\{-c/b\}$)
 else if $\text{discr} \geq 0$ then $\{(-b + \text{sqrt } \text{discr}) / (2 * a), (-b - \text{sqrt } \text{discr}) / (2 * a)\}$
 else {}) (is ?th1)
 ⟨proof⟩

lemma *finite-quadratic-equation-solutions-reals*:
fixes $a \ b \ c :: \text{real}$
defines $\text{discr} \equiv (b^2 - 4 * a * c)$
shows $\text{finite } \{x. a * x^2 + b * x + c = 0\} \longleftrightarrow a \neq 0 \vee b \neq 0 \vee c \neq 0$
 ⟨proof⟩

lemma *card-quadratic-equation-solutions-reals*:
fixes $a \ b \ c :: \text{real}$
defines $\text{discr} \equiv (b^2 - 4 * a * c)$

shows $\text{card } \{x. a * x^2 + b * x + c = 0\} =$
 (if $a = 0$ then
 (if $b = 0$ then 0 else 1)
 else if $\text{discr} \geq 0$ then if $\text{discr} = 0$ then 1 else 2 else 0) (is ?th1)

<proof>

lemma *card-quadratic-equation-solutions-reals-le-2*:
 $\text{card } \{x :: \text{real}. a * x^2 + b * x + c = 0\} \leq 2$
<proof>

lemma *quadratic-equation-solution-rat-iff*:
fixes $a b c :: \text{int}$ **and** $x y :: \text{real}$
defines $f \equiv (\lambda x :: \text{real}. a * x^2 + b * x + c)$
defines $\text{discr} \equiv \text{nat } (b^2 - 4 * a * c)$
assumes $a \neq 0 \wedge f x = 0$
shows $x \in \mathbb{Q} \longleftrightarrow \text{is-square } \text{discr}$
<proof>

2.4 Periodic continued fractions and quadratic irrationals

We now show the main result: A positive irrational number has a periodic continued fraction expansion iff it is a quadratic irrational.

In principle, this statement naturally also holds for negative numbers, but the current formalisation of continued fractions only supports non-negative numbers. It also holds for rational numbers in some sense, since their continued fraction expansion is finite to begin with.

theorem *periodic-cfrac-imp-quadratic-irrational*:
assumes [*simp*]: $\text{cfrac-length } c = \infty$
and *period*: $l > 0 \wedge k \geq N \implies \text{cfrac-nth } c (k + l) = \text{cfrac-nth } c k$
shows *quadratic-irrational* ($\text{cfrac-lim } c$)
<proof>

lift-definition *pproduct-cfrac* :: $\text{nat list} \Rightarrow \text{cfrac}$ **is**
 $\lambda xs. \text{if } xs = [] \text{ then } (0, \text{LNil}) \text{ else}$
 $(\text{int } (\text{hd } xs), \text{lstream-of-stream } (\text{cycle } (\text{map } (\lambda n. n - 1) (\text{tl } xs @ [\text{hd } xs])))$
<proof>

definition *periodic-cfrac* :: $\text{int list} \Rightarrow \text{int list} \Rightarrow \text{cfrac}$ **where**
 $\text{periodic-cfrac } xs \ ys = \text{cfrac-of-stream } (\text{Stream.shift } xs (\text{Stream.cycle } ys))$

lemma *periodic-cfrac-Nil* [*simp*]: $\text{pproduct-cfrac } [] = 0$
<proof>

lemma *cfrac-length-pproduct-cfrac* [*simp*]:
 $xs \neq [] \implies \text{cfrac-length } (\text{pproduct-cfrac } xs) = \infty$
<proof>

lemma *cfrac-nth-pproperiodic-cfrac*:
assumes $xs \neq []$ **and** $0 \notin \text{set } xs$
shows $\text{cfrac-nth } (\text{pproperiodic-cfrac } xs) \ n = xs \ ! \ (n \ \text{mod} \ \text{length } xs)$
 $\langle \text{proof} \rangle$

definition *pproperiodic-cfrac-info* :: $\text{nat list} \Rightarrow \text{int} \times \text{int} \times \text{int}$ **where**
pproperiodic-cfrac-info $xs =$
 $(\text{let } l = \text{length } xs;$
 $h = \text{conv-num-fun } (\lambda n. xs \ ! \ n);$
 $k = \text{conv-denom-fun } (\lambda n. xs \ ! \ n);$
 $A = k \ (l - 1);$
 $B = h \ (l - 1) - (\text{if } l = 1 \ \text{then } 0 \ \text{else } k \ (l - 2));$
 $C = (\text{if } l = 1 \ \text{then } -1 \ \text{else } -h \ (l - 2))$
 $\text{in } (B^{\wedge}2 - 4 * A * C, B, 2 * A)$

lemma *conv-gen-cong*:
assumes $\forall k \in \{n..N\}. f \ k = f' \ k$
shows $\text{conv-gen } f \ (a, b, n) \ N = \text{conv-gen } f' \ (a, b, n) \ N$
 $\langle \text{proof} \rangle$

lemma
assumes $\forall k \leq n. c \ k = \text{cfrac-nth } c' \ k$
shows $\text{conv-num-fun-eq}' : \text{conv-num-fun } c \ n = \text{conv-num } c' \ n$
and $\text{conv-denom-fun-eq}' : \text{conv-denom-fun } c \ n = \text{conv-denom } c' \ n$
 $\langle \text{proof} \rangle$

lemma *gcd-minus-commute-left*: $\text{gcd } (a - b :: 'a :: \text{ring-gcd}) \ c = \text{gcd } (b - a) \ c$
 $\langle \text{proof} \rangle$

lemma *gcd-minus-commute-right*: $\text{gcd } c \ (a - b :: 'a :: \text{ring-gcd}) = \text{gcd } c \ (b - a)$
 $\langle \text{proof} \rangle$

lemma *periodic-cfrac-info-aux*:
fixes $D \ E \ F :: \text{int}$
assumes $\text{pproperiodic-cfrac-info } xs = (D, E, F)$
assumes $xs \neq []$ $0 \notin \text{set } xs$
shows $\text{cfrac-lim } (\text{pproperiodic-cfrac } xs) = (\text{sqrt } D + E) / F$
and $D > 0$ **and** $F > 0$
 $\langle \text{proof} \rangle$

We can now compute surd representations for (purely) periodic continued fractions, e.g. $[1, 1, 1, \dots] = \frac{\sqrt{5}+1}{2}$:

value *pproperiodic-cfrac-info* [1]

We can now compute surd representations for periodic continued fractions, e.g. $[1, 1, 1, 1, 6] = \frac{\sqrt{13}+3}{4}$:

value *pproperiodic-cfrac-info* [1,1,1,1,6]

With a little bit of work, one could also easily derive from this a version for non-purely periodic continued fraction.

Next, we show that any quadratic irrational has a periodic continued fraction expansion.

theorem *quadratic-irrational-imp-periodic-cfrac:*

assumes *quadratic-irrational* (*cfrac-lim* *e*)

obtains $N\ l$ **where** $l > 0$ **and** $\bigwedge n\ m.\ n \geq N \implies \text{cfrac-nth } e\ (n + m * l) = \text{cfrac-nth } e\ n$

and *cfrac-remainder* $e\ (N + l) = \text{cfrac-remainder } e\ N$

and *cfrac-length* $e = \infty$

<proof>

theorem *periodic-cfrac-iff-quadratic-irrational:*

assumes $x \notin \mathbb{Q}$ $x \geq 0$

shows *quadratic-irrational* $x \longleftrightarrow$

$(\exists N\ l.\ l > 0 \wedge (\forall n \geq N.\ \text{cfrac-nth } (\text{cfrac-of-real } x)\ (n + l) = \text{cfrac-nth } (\text{cfrac-of-real } x)\ n))$

<proof>

The following result can e.g. be used to show that a number is *not* a quadratic irrational.

lemma *quadratic-irrational-cfrac-nth-range-finite:*

assumes *quadratic-irrational* (*cfrac-lim* *e*)

shows *finite* (*range* (*cfrac-nth* *e*))

<proof>

end

3 The continued fraction expansion of e

theory *E-CFrac*

imports

HOL-Analysis.Analysis

Continued-Fractions

Quadratic-Irrationals

begin

lemma *fact-real-at-top: filterlim* (*fact* :: *nat* \implies *real*) *at-top at-top*

<proof>

lemma *filterlim-div-nat-at-top:*

assumes *filterlim* *f* *at-top* $F\ m > 0$

shows *filterlim* $(\lambda x.\ f\ x\ \text{div } m :: \text{nat})$ *at-top* F

<proof>

The continued fraction expansion of e has the form $[2; 1, 2, 1, 1, 4, 1, 1, 6, 1, 1, 8, 1, \dots]$:

definition *e-cfrac* **where**

$e\text{-frac} = \text{frac } (\lambda n. \text{if } n = 0 \text{ then } 2 \text{ else if } n \bmod 3 = 2 \text{ then } 2 * (\text{Suc } n \text{ div } 3) \text{ else } 1)$

lemma *cfrac-nth-e*:

$c\text{frac-nth } e\text{-frac } n = (\text{if } n = 0 \text{ then } 2 \text{ else if } n \bmod 3 = 2 \text{ then } 2 * (\text{Suc } n \text{ div } 3) \text{ else } 1)$
 $\langle \text{proof} \rangle$

lemma *cfrac-length-e [simp]*: $c\text{frac-length } e\text{-frac} = \infty$

$\langle \text{proof} \rangle$

The formalised proof follows the one from Proof Wiki [2].

context

fixes $A B C :: \text{nat} \Rightarrow \text{real}$ **and** $p q :: \text{nat} \Rightarrow \text{int}$ **and** $a :: \text{nat} \Rightarrow \text{int}$

defines $A \equiv (\lambda n. \text{integral } \{0..1\} (\lambda x. \text{exp } x * x^n * (x - 1)^n / \text{fact } n))$

and $B \equiv (\lambda n. \text{integral } \{0..1\} (\lambda x. \text{exp } x * x^{\text{Suc } n} * (x - 1)^n / \text{fact } n))$

and $C \equiv (\lambda n. \text{integral } \{0..1\} (\lambda x. \text{exp } x * x^n * (x - 1)^{\text{Suc } n} / \text{fact } n))$

and $p \equiv (\lambda n. \text{if } n \leq 1 \text{ then } 1 \text{ else conv-num } e\text{-frac } (n - 2))$

and $q \equiv (\lambda n. \text{if } n = 0 \text{ then } 1 \text{ else if } n = 1 \text{ then } 0 \text{ else conv-denom } e\text{-frac } (n - 2))$

and $a \equiv (\lambda n. \text{if } n \bmod 3 = 2 \text{ then } 2 * (\text{Suc } n \text{ div } 3) \text{ else } 1)$

begin

lemma

assumes $n \geq 2$

shows $p\text{-rec}: p\ n = a\ (n - 2) * p\ (n - 1) + p\ (n - 2)$ (**is ?th1**)

and $q\text{-rec}: q\ n = a\ (n - 2) * q\ (n - 1) + q\ (n - 2)$ (**is ?th2**)

$\langle \text{proof} \rangle$

lemma

assumes $n \geq 1$

shows $p\text{-rec0}: p\ (3 * n) = p\ (3 * n - 1) + p\ (3 * n - 2)$

and $q\text{-rec0}: q\ (3 * n) = q\ (3 * n - 1) + q\ (3 * n - 2)$

$\langle \text{proof} \rangle$

lemma

assumes $n \geq 1$

shows $p\text{-rec1}: p\ (3 * n + 1) = 2 * \text{int } n * p\ (3 * n) + p\ (3 * n - 1)$

and $q\text{-rec1}: q\ (3 * n + 1) = 2 * \text{int } n * q\ (3 * n) + q\ (3 * n - 1)$

$\langle \text{proof} \rangle$

lemma $p\text{-rec2}: p\ (3 * n + 2) = p\ (3 * n + 1) + p\ (3 * n)$

and $q\text{-rec2}: q\ (3 * n + 2) = q\ (3 * n + 1) + q\ (3 * n)$

$\langle \text{proof} \rangle$

lemma $A\text{-0}: A\ 0 = \text{exp } 1 - 1$ **and** $B\text{-0}: B\ 0 = 1$ **and** $C\text{-0}: C\ 0 = 2 - \text{exp } 1$

$\langle \text{proof} \rangle$

lemma $A\text{-bound}: \text{norm } (A\ n) \leq \text{exp } 1 / \text{fact } n$

<proof>

lemma *B-bound*: $\text{norm } (B\ n) \leq \text{exp } 1 / \text{fact } n$
<proof>

lemma *C-bound*: $\text{norm } (C\ n) \leq \text{exp } 1 / \text{fact } n$
<proof>

lemma *A-Suc*: $A\ (\text{Suc } n) = -B\ n - C\ n$
<proof>

lemma *B-Suc*: $B\ (\text{Suc } n) = -2 * \text{Suc } n * A\ (\text{Suc } n) + C\ n$
<proof>

lemma *C-Suc*: $C\ n = B\ n - A\ n$
<proof>

lemma *unfold-add-numeral*: $c * n + \text{numeral } b = \text{Suc } (c * n + \text{pred-numeral } b)$
<proof>

lemma *ABC*:
 $A\ n = q\ (\mathcal{I} * n) * \text{exp } 1 - p\ (\mathcal{I} * n) \wedge$
 $B\ n = p\ (\text{Suc } (\mathcal{I} * n)) - q\ (\text{Suc } (\mathcal{I} * n)) * \text{exp } 1 \wedge$
 $C\ n = p\ (\text{Suc } (\text{Suc } (\mathcal{I} * n))) - q\ (\text{Suc } (\text{Suc } (\mathcal{I} * n))) * \text{exp } 1$
<proof>

lemma *q-pos*: $q\ n > 0$ **if** $n \neq 1$
<proof>

lemma *conv-diff-exp-bound*: $\text{norm } (\text{exp } 1 - p\ n / q\ n) \leq \text{exp } 1 / \text{fact } (n\ \text{div } 3)$
<proof>

theorem *e-cfrac*: $\text{cfrac-lim } e\text{-cfrac} = \text{exp } 1$
<proof>

corollary *e-cfrac-altdef*: $e\text{-cfrac} = \text{cfrac-of-real } (\text{exp } 1)$
<proof>

This also provides us with a nice proof that e is not rational and not a quadratic irrational either.

corollary *exp1-irrational*: $(\text{exp } 1 :: \text{real}) \notin \mathbb{Q}$
<proof>

corollary *exp1-not-quadratic-irrational*: $\neg \text{quadratic-irrational } (\text{exp } 1 :: \text{real})$
<proof>

end
end

4 Continued fraction expansions for square roots of naturals

```

theory Sqrt-Nat-Cfrac
imports
  Quadratic-Irrationals
  HOL-Library.While-Combinator
  HOL-Library.IArray
begin

```

In this section, we shall explore the continued fraction expansion of \sqrt{D} , where D is a natural number.

lemma *butlast-nth* [simp]: $n < \text{length } xs - 1 \implies \text{butlast } xs ! n = xs ! n$
 ⟨proof⟩

The following is the length of the period in the continued fraction expansion of \sqrt{D} for a natural number D .

definition *sqrt-nat-period-length* :: $\text{nat} \Rightarrow \text{nat}$ **where**
sqrt-nat-period-length $D =$
 (if *is-square* D then 0
 else (LEAST l . $l > 0 \wedge (\forall n. \text{cfrac-nth } (\text{cfrac-of-real } (\text{sqrt } D)) (\text{Suc } n + l) = \text{cfrac-nth } (\text{cfrac-of-real } (\text{sqrt } D)) (\text{Suc } n))$))

Next, we define a more workable representation for the continued fraction expansion of \sqrt{D} consisting of the period length, the natural number $\lfloor \sqrt{D} \rfloor$, and the content of the period.

definition *sqrt-cfrac-info* :: $\text{nat} \Rightarrow \text{nat} \times \text{nat} \times \text{nat list}$ **where**
sqrt-cfrac-info $D =$
 (*sqrt-nat-period-length* D , *floor-sqrt* D ,
 map ($\lambda n. \text{nat } (\text{cfrac-nth } (\text{cfrac-of-real } (\text{sqrt } D)) (\text{Suc } n))$) [0..*sqrt-nat-period-length* D])

lemma *sqrt-nat-period-length-square* [simp]: *is-square* $D \implies \text{sqrt-nat-period-length } D = 0$
 ⟨proof⟩

definition *sqrt-cfrac* :: $\text{nat} \Rightarrow \text{cfrac}$
where *sqrt-cfrac* $D = \text{cfrac-of-real } (\text{sqrt } (\text{real } D))$

```

context
  fixes  $D D' :: \text{nat}$ 
  defines  $D' \equiv \text{nat } \lfloor \text{sqrt } D \rfloor$ 
begin

```

A number $\alpha = \frac{\sqrt{D}+p}{q}$ for $p, q \in \mathbb{N}$ is called a *reduced quadratic surd* if $\alpha > 1$ and $\bar{\alpha} \in (-1; 0)$, where $\bar{\alpha}$ denotes the conjugate $\frac{-\sqrt{D}+p}{q}$. It is furthermore called *associated* to D if q divides $D - p^2$.

definition *red-assoc* :: $\text{nat} \times \text{nat} \Rightarrow \text{bool}$ **where**

red-assoc = $(\lambda(p, q).$
 $q > 0 \wedge q \text{ dvd } (D - p^2) \wedge (\text{sqrt } D + p) / q > 1 \wedge (-\text{sqrt } D + p) / q \in$
 $\{-1 < .. < 0\})$

The following two functions convert between a surd represented as a pair of natural numbers and the actual real number and its conjugate:

definition *surd-to-real* :: $\text{nat} \times \text{nat} \Rightarrow \text{real}$

where *surd-to-real* = $(\lambda(p, q). (\text{sqrt } D + p) / q)$

definition *surd-to-real-cnj* :: $\text{nat} \times \text{nat} \Rightarrow \text{real}$

where *surd-to-real-cnj* = $(\lambda(p, q). (-\text{sqrt } D + p) / q)$

The next function performs a single step in the continued fraction expansion of \sqrt{D} .

definition *sqrt-remainder-step* :: $\text{nat} \times \text{nat} \Rightarrow \text{nat} \times \text{nat}$ **where**

sqrt-remainder-step = $(\lambda(p, q). \text{let } X = (p + D') \text{ div } q; p' = X * q - p \text{ in } (p',$
 $(D - p'^2) \text{ div } q))$

If we iterate this step function starting from the surd $\frac{1}{\sqrt{D} - \lfloor \sqrt{D} \rfloor}$, we get the entire expansion.

definition *sqrt-remainder-surd* :: $\text{nat} \Rightarrow \text{nat} \times \text{nat}$

where *sqrt-remainder-surd* = $(\lambda n. (\text{sqrt-remainder-step} \hat{\sim} n) (D', D - D'^2))$

context

fixes *sqrt-cfrac-nth* :: $\text{nat} \Rightarrow \text{nat}$ **and** *l*

assumes *nonsquare*: $\neg \text{is-square } D$

defines *sqrt-cfrac-nth* $\equiv (\lambda n. \text{case } \text{sqrt-remainder-surd } n \text{ of } (p, q) \Rightarrow (D' + p)$
 $\text{div } q)$

defines *l* $\equiv \text{sqrt-nat-period-length } D$

begin

lemma *D'-pos*: $D' > 0$

<proof>

lemma *D'-sqr-less-D*: $D'^2 < D$

<proof>

lemma *red-assoc-imp-irrat*:

assumes *red-assoc* *pq*

shows *surd-to-real* *pq* $\notin \mathbb{Q}$

<proof>

lemma *surd-to-real-cnj-irrat*:

assumes *red-assoc* *pq*

shows *surd-to-real-cnj* *pq* $\notin \mathbb{Q}$

<proof>

lemma *surd-to-real-nonneg* [intro]: *surd-to-real* $pq \geq 0$
 ⟨proof⟩

lemma *surd-to-real-pos* [intro]: *red-assoc* $pq \implies \textit{surd-to-real}$ $pq > 0$
 ⟨proof⟩

lemma *surd-to-real-nz* [simp]: *red-assoc* $pq \implies \textit{surd-to-real}$ $pq \neq 0$
 ⟨proof⟩

lemma *surd-to-real-cnj-nz* [simp]: *red-assoc* $pq \implies \textit{surd-to-real-cnj}$ $pq \neq 0$
 ⟨proof⟩

lemma *red-assoc-step*:

assumes *red-assoc* pq

defines $X \equiv (D' + \textit{fst}$ $pq) \textit{div}$ \textit{snd} pq

defines $pq' \equiv \textit{sqrt-remainder-step}$ pq

shows *red-assoc* pq'

$\textit{surd-to-real}$ $pq' = 1 / \textit{frac}$ (*surd-to-real* pq)

$\textit{surd-to-real-cnj}$ $pq' = 1 / (\textit{surd-to-real-cnj}$ $pq - X)$

$X > 0$ $X * \textit{snd}$ $pq \leq 2 * D' X = \textit{nat}$ [*surd-to-real* pq]

$X = \textit{nat}$ [$-1 / \textit{surd-to-real-cnj}$ pq']

⟨proof⟩

lemma *red-assoc-denom-2D*:

assumes *red-assoc* (p, q)

defines $X \equiv (D' + p) \textit{div}$ q

assumes $X > D'$

shows $q = 1$

⟨proof⟩

lemma *red-assoc-denom-1*:

assumes *red-assoc* $(p, 1)$

shows $p = D'$

⟨proof⟩

lemma *red-assoc-begin*:

red-assoc $(D', D - D^2)$

$\textit{surd-to-real}$ $(D', D - D^2) = 1 / \textit{frac}$ (*sqrt* D)

$\textit{surd-to-real-cnj}$ $(D', D - D^2) = -1 / (\textit{sqrt}$ $D + D')$

⟨proof⟩

lemma *cfrac-remainder-surd-to-real*:

assumes *red-assoc* pq

shows *cfrac-remainder* (*cfrac-of-real* (*surd-to-real* pq)) $n =$

$\textit{surd-to-real}$ (*sqrt-remainder-step* $\widehat{\sim} n$) pq)

⟨proof⟩

lemma *red-assoc-step'* [intro]: *red-assoc* $pq \implies \textit{red-assoc}$ (*sqrt-remainder-step* pq)

⟨proof⟩

lemma *red-assoc-steps* [intro]: $red\text{-}assoc\ pq \implies red\text{-}assoc ((sqrt\text{-}remainder\text{-}step \sim n) pq)$
 ⟨proof⟩

lemma *floor-sqrt-less-sqrt*: $D' < sqrt\ D$
 ⟨proof⟩

lemma *red-assoc-bounds*:
 assumes *red-assoc* pq
 shows $pq \in (SIGMA\ p:\{0 < .. D'\}, \{Suc\ D' - p .. D' + p\})$
 ⟨proof⟩

lemma *surd-to-real-cnj-eq-iff*:
 assumes *red-assoc* pq *red-assoc* pq'
 shows $surd\text{-}to\text{-}real\text{-}cnj\ pq = surd\text{-}to\text{-}real\text{-}cnj\ pq' \longleftrightarrow pq = pq'$
 ⟨proof⟩

lemma *red-assoc-sqrt-remainder-surd* [intro]: $red\text{-}assoc (sqrt\text{-}remainder\text{-}surd\ n)$
 ⟨proof⟩

lemma *surd-to-real-sqrt-remainder-surd*:
 $surd\text{-}to\text{-}real (sqrt\text{-}remainder\text{-}surd\ n) = cfrac\text{-}remainder (cfrac\text{-}of\text{-}real (sqrt\ D))$
 (Suc n)
 ⟨proof⟩

lemma *sqrt-cfrac*: $sqrt\text{-}cfrac\text{-}nth\ n = cfrac\text{-}nth (cfrac\text{-}of\text{-}real (sqrt\ D)) (Suc\ n)$
 ⟨proof⟩

lemma *sqrt-cfrac-pos*: $sqrt\text{-}cfrac\text{-}nth\ k > 0$
 ⟨proof⟩

lemma *snd-sqrt-remainder-surd-pos*: $snd (sqrt\text{-}remainder\text{-}surd\ n) > 0$
 ⟨proof⟩

lemma
 shows *period-nonempty*: $l > 0$
 and *period-length-le-aux*: $l \leq D' * (D' + 1)$
 and *sqrt-remainder-surd-periodic*: $\bigwedge n. sqrt\text{-}remainder\text{-}surd\ n = sqrt\text{-}remainder\text{-}surd\ (n\ mod\ l)$
 and *sqrt-cfrac-periodic*: $\bigwedge n. sqrt\text{-}cfrac\text{-}nth\ n = sqrt\text{-}cfrac\text{-}nth\ (n\ mod\ l)$
 and *sqrt-remainder-surd-smallest-period*:
 $\bigwedge n. n \in \{0 < .. < l\} \implies sqrt\text{-}remainder\text{-}surd\ n \neq sqrt\text{-}remainder\text{-}surd\ 0$
 and *snd-sqrt-remainder-surd-gt-1*: $\bigwedge n. n < l - 1 \implies snd (sqrt\text{-}remainder\text{-}surd\ n) > 1$
 and *sqrt-cfrac-le*: $\bigwedge n. n < l - 1 \implies sqrt\text{-}cfrac\text{-}nth\ n \leq D'$
 and *sqrt-remainder-surd-last*: $sqrt\text{-}remainder\text{-}surd\ (l - 1) = (D', 1)$
 and *sqrt-cfrac-last*: $sqrt\text{-}cfrac\text{-}nth\ (l - 1) = 2 * D'$

and *sqrt-cfrac-palindrome*: $\bigwedge n. n < l - 1 \implies \text{sqrt-cfrac-nth } (l - n - 2) = \text{sqrt-cfrac-nth } n$

and *sqrt-cfrac-smallest-period*:
 $\bigwedge l'. l' > 0 \implies (\bigwedge k. \text{sqrt-cfrac-nth } (k + l') = \text{sqrt-cfrac-nth } k) \implies l' \geq l$
 $\langle \text{proof} \rangle$

theorem *cfrac-sqrt-periodic*:
 $\text{cfrac-nth } (\text{cfrac-of-real } (\text{sqrt } D)) (\text{Suc } n) =$
 $\text{cfrac-nth } (\text{cfrac-of-real } (\text{sqrt } D)) (\text{Suc } (n \bmod l))$
 $\langle \text{proof} \rangle$

theorem *cfrac-sqrt-le*: $n \in \{0 < .. < l\} \implies \text{cfrac-nth } (\text{cfrac-of-real } (\text{sqrt } D)) n \leq D'$
 $\langle \text{proof} \rangle$

theorem *cfrac-sqrt-last*: $\text{cfrac-nth } (\text{cfrac-of-real } (\text{sqrt } D)) l = 2 * D'$
 $\langle \text{proof} \rangle$

theorem *cfrac-sqrt-palindrome*:
assumes $n \in \{0 < .. < l\}$
shows $\text{cfrac-nth } (\text{cfrac-of-real } (\text{sqrt } D)) (l - n) = \text{cfrac-nth } (\text{cfrac-of-real } (\text{sqrt } D)) n$
 $\langle \text{proof} \rangle$

lemma *sqrt-cfrac-info-palindrome*:
assumes $\text{sqrt-cfrac-info } D = (a, b, cs)$
shows $\text{rev } (\text{butlast } cs) = \text{butlast } cs$
 $\langle \text{proof} \rangle$

lemma *sqrt-cfrac-info-last*:
assumes $\text{sqrt-cfrac-info } D = (a, b, cs)$
shows $\text{last } cs = 2 * \text{floor-sqrt } D$
 $\langle \text{proof} \rangle$

The following lemmas allow us to compute the period of the expansion of the square root:

lemma *while-option-sqrt-cfrac*:
defines $\text{step}' \equiv (\lambda(as, pq). ((D' + \text{fst } pq) \text{ div } \text{snd } pq \# as, \text{sqrt-remainder-step } pq))$
defines $b \equiv (\lambda(-, pq). \text{snd } pq \neq 1)$
defines $\text{initial} \equiv ([\] :: \text{nat list}, (D', D - D^2))$
shows $\text{while-option } b \text{ step}' \text{ initial} =$
 $\text{Some } (\text{rev } (\text{map } \text{sqrt-cfrac-nth } [0..<l - 1]), (D', 1))$
 $\langle \text{proof} \rangle$

lemma *while-option-sqrt-cfrac-info*:
defines $\text{step}' \equiv (\lambda(as, pq). ((D' + \text{fst } pq) \text{ div } \text{snd } pq \# as, \text{sqrt-remainder-step } pq))$
defines $b \equiv (\lambda(-, pq). \text{snd } pq \neq 1)$
defines $\text{initial} \equiv ([\], (D', D - D^2))$

shows *sqrt-cfrac-info* $D =$
 (case while-option b step' initial of
 *Some (as, -) ⇒ (Suc (length as), D', rev ((2 * D') # as)))*
 ⟨proof⟩

end
end

lemma *sqrt-nat-period-length-le*: *sqrt-nat-period-length* $D \leq \text{nat } \lfloor \text{sqrt } D \rfloor * (\text{nat } \lfloor \text{sqrt } D \rfloor + 1)$
 ⟨proof⟩

lemma *sqrt-nat-period-length-0-iff* [simp]:
sqrt-nat-period-length $D = 0 \iff \text{is-square } D$
 ⟨proof⟩

lemma *sqrt-nat-period-length-pos-iff* [simp]:
sqrt-nat-period-length $D > 0 \iff \neg \text{is-square } D$
 ⟨proof⟩

lemma *sqrt-cfrac-info-code* [code]:
sqrt-cfrac-info $D =$
 (let D' = floor-sqrt D
 in if D² = D then (0, D', [])
 else
 case while-option
 (λ(-, pq). snd pq ≠ 1)
 *(λ(as, (p, q)). let X = (p + D') div q; p' = X * q - p*
 in (X # as, p', (D - p²) div q))
 ([], D', D - D²)
 *of Some (as, -) ⇒ (Suc (length as), D', rev ((2 * D') # as))*)
 ⟨proof⟩

lemma *sqrt-nat-period-length-code* [code]:
sqrt-nat-period-length $D = \text{fst } (\text{sqrt-cfrac-info } D)$
 ⟨proof⟩

For efficiency reasons, it is often better to use an array instead of a list:

definition *sqrt-cfrac-info-array* **where**
sqrt-cfrac-info-array $D = (\text{case } \text{sqrt-cfrac-info } D \text{ of } (a, b, c) \Rightarrow (a, b, \text{IArray } c))$

lemma *fst-sqrt-cfrac-info-array* [simp]: *fst* (*sqrt-cfrac-info-array* D) = *sqrt-nat-period-length* D
 ⟨proof⟩

lemma *snd-sqrt-cfrac-info-array* [simp]: *fst* (*snd* (*sqrt-cfrac-info-array* D)) = *floor-sqrt* D
 ⟨proof⟩

definition *cfrac-sqrt-nth* :: $\text{nat} \times \text{nat} \times \text{nat iarray} \Rightarrow \text{nat} \Rightarrow \text{nat}$ **where**
cfrac-sqrt-nth info n =
(case info of (l, a0, as) \Rightarrow if $n = 0$ then $a0$ else $as !! ((n - 1) \bmod l)$)

lemma *cfrac-sqrt-nth*:
assumes $\neg \text{is-square } D$
shows *cfrac-nth (cfrac-of-real (sqrt D)) n =*
int (cfrac-sqrt-nth (sqrt-cfrac-info-array D) n) (is ?lhs = ?rhs)
<proof>

lemma *sqrt-cfrac-code* [*code*]:
sqrt-cfrac D =
(let info = sqrt-cfrac-info-array D;
(l, a0, -) = info
in if $l = 0$ then cfrac-of-int (int a0) else cfrac (cfrac-sqrt-nth info))
<proof>

As a test, we determine the continued fraction expansion of $\sqrt{129}$, which is $[11; \overline{2, 1, 3, 1, 6, 1, 3, 1, 2, 22}]$ (a period length of 10):

value *let info = sqrt-cfrac-info-array 129 in info*
value *sqrt-nat-period-length 129*

We can also compute convergents of $\sqrt{129}$ and observe that the difference between the square of the convergents and 129 vanishes quickly::

value *map (conv (sqrt-cfrac 129)) [0..<10]*
value *map ($\lambda n. |conv (sqrt-cfrac 129) n|^2 - 129|$) [0..<20]*

end

5 Lifting solutions of Pell's Equation

theory *Pell-Lifting*
imports *Pell.Pell Pell.Pell-Algorithm*
begin

5.1 Auxiliary material

lemma (in *pell*) *snth-pell-solutions*: *snth (pell-solutions D) n = nth-solution n*
<proof>

definition *square-squarefree-part-nat* :: $\text{nat} \Rightarrow \text{nat} \times \text{nat}$ **where**
square-squarefree-part-nat n = (square-part n, squarefree-part n)

lemma *prime-factorization-squarefree-part*:
assumes $x \neq 0$
shows *prime-factorization (squarefree-part x) =*
mset-set { $p \in \text{prime-factors } x. \text{ odd (multiplicity } p \ x)$ } (is ?lhs = ?rhs)

<proof>

lemma *squarefree-part-nat*:

squarefree-part ($n :: \text{nat}$) = $(\prod \{p \in \text{prime-factors } n. \text{odd } (\text{multiplicity } p \ n)\})$

<proof>

lemma *prime-factorization-square-part*:

assumes $x \neq 0$

shows *prime-factorization* (*square-part* x) =

$(\sum p \in \text{prime-factors } x. \text{replicate-mset } (\text{multiplicity } p \ x \ \text{div } 2) \ p)$ (**is** *?lhs*)

<proof>

lemma *prod-mset-sum*: *prod-mset* (*sum* $f \ A$) = $(\prod x \in A. \text{prod-mset } (f \ x))$

<proof>

lemma *square-part-nat*:

assumes $n > 0$

shows *square-part* ($n :: \text{nat}$) = $(\prod p \in \text{prime-factors } n. p \wedge (\text{multiplicity } p \ n \ \text{div } 2))$

<proof>

lemma *square-squarefree-part-nat-code* [*code*]:

square-squarefree-part-nat n = (*if* $n = 0$ *then* $(0, 1)$

else let $ps = \text{prime-factorization } n$

in $(\prod p \in \text{set-mset } ps. p \wedge (\text{count } ps \ p \ \text{div } 2))$,

$\prod (\text{Set.filter } (\lambda p. \text{odd } (\text{count } ps \ p)) (\text{set-mset } ps))$)

<proof>

lemma *square-part-nat-code* [*code-unfold*]:

square-part ($n :: \text{nat}$) = (*if* $n = 0$ *then* 0

else let $ps = \text{prime-factorization } n$ *in* $(\prod p \in \text{set-mset } ps. p \wedge (\text{count } ps \ p \ \text{div } 2))$)

<proof>

lemma *squarefree-part-nat-code* [*code-unfold*]:

squarefree-part ($n :: \text{nat}$) = (*if* $n = 0$ *then* 1

else let $ps = \text{prime-factorization } n$ *in* $(\prod (\text{Set.filter } (\lambda p. \text{odd } (\text{count } ps \ p)) (\text{set-mset } ps)))$)

<proof>

lemma *is-nth-power-mult-nth-powerD*:

assumes *is-nth-power* n ($a * b \wedge n$) $b > 0 \ n > 0$

shows *is-nth-power* n ($a :: \text{nat}$)

<proof>

lemma (**in** *pell*) *fund-sol-eq-fstI*:

assumes *nontriv-solution* (x, y)

assumes $\wedge x' \ y'. \text{nontriv-solution } (x', y') \implies x \leq x'$

shows $\text{fund-sol} = (x, y)$
 $\langle \text{proof} \rangle$

lemma (in *pell*) *fund-sol-eqI-fst'*:
assumes *nontriv-solution* xy
assumes $\bigwedge x' y'. \text{nontriv-solution } (x', y') \implies \text{fst } xy \leq x'$
shows $\text{fund-sol} = xy$
 $\langle \text{proof} \rangle$

lemma (in *pell*) *fund-sol-eq-sndI*:
assumes *nontriv-solution* (x, y)
assumes $\bigwedge x' y'. \text{nontriv-solution } (x', y') \implies y \leq y'$
shows $\text{fund-sol} = (x, y)$
 $\langle \text{proof} \rangle$

lemma (in *pell*) *fund-sol-eqI-snd'*:
assumes *nontriv-solution* xy
assumes $\bigwedge x' y'. \text{nontriv-solution } (x', y') \implies \text{snd } xy \leq y'$
shows $\text{fund-sol} = xy$
 $\langle \text{proof} \rangle$

5.2 The lifting mechanism

The solutions of Pell's equations for parameters D and $a^2 D$ stand in correspondence to one another: every solution (x, y) for parameter D can be lowered to a solution (x, ay) for $a^2 D$, and every solution of the form (x, ay) for parameter $a^2 D$ can be lifted to a solution (x, y) for parameter D .

locale *pell-lift* = *pell* +
fixes $a D' :: \text{nat}$
assumes $\text{nz}: a > 0$
defines $D' \equiv D * a^2$
begin

lemma *nonsquare-D'*: $\neg \text{is-square } D'$
 $\langle \text{proof} \rangle$

definition *lift-solution* :: $\text{nat} \times \text{nat} \Rightarrow \text{nat} \times \text{nat}$ **where**
 $\text{lift-solution} = (\lambda(x, y). (x, y \text{ div } a))$

definition *lower-solution* :: $\text{nat} \times \text{nat} \Rightarrow \text{nat} \times \text{nat}$ **where**
 $\text{lower-solution} = (\lambda(x, y). (x, y * a))$

definition *liftable-solution* :: $\text{nat} \times \text{nat} \Rightarrow \text{bool}$ **where**
 $\text{liftable-solution} = (\lambda(x, y). a \text{ dvd } y)$

sublocale *lift*: *pell* D'
 $\langle \text{proof} \rangle$

lemma *lift-solution-iff*: $\text{lift.solution } xy \longleftrightarrow \text{solution } (\text{lower-solution } xy)$
 ⟨proof⟩

lemma *lift-solution*:
assumes $\text{solution } xy \text{ liftable-solution } xy$
shows $\text{lift.solution } (\text{lift-solution } xy)$
 ⟨proof⟩

In particular, the fundamental solution for $a^2 D$ is the smallest liftable solution for D :

lemma *lift-fund-sol*:
assumes $\bigwedge n. 0 < n \implies n < m \implies \neg \text{liftable-solution } (\text{nth-solution } n)$
assumes $\text{liftable-solution } (\text{nth-solution } m) \ m > 0$
shows $\text{lift.fund-sol} = \text{lift-solution } (\text{nth-solution } m)$
 ⟨proof⟩

end

5.3 Accelerated computation of the fundamental solution for non-squarefree inputs

Solving Pell's equation for some D of the form $a^2 D'$ can be done by solving it for D' and then lifting the solution. Thus, if D is not squarefree, we can compute its squarefree decomposition $a^2 D'$ with D' squarefree and thus speed up the computation (since D' is smaller than D).

The squarefree decomposition can only be computed (according to current knowledge in mathematics) through the prime decomposition. However, given how big the solutions are for even moderate values of D , it is usually worth doing it if D is not squarefree.

lemma *squarefree-part-of-square* [*simp*]:
assumes $\text{is-square } (x :: 'a :: \{\text{factorial-semiring, normalization-semidom-multiplicative}\})$
assumes $x \neq 0$
shows $\text{squarefree-part } x = \text{unit-factor } x$
 ⟨proof⟩

lemma *squarefree-part-1-imp-square*:
assumes $\text{squarefree-part } x = 1$
shows $\text{is-square } x$
 ⟨proof⟩

definition *find-fund-sol-fast* **where**
 $\text{find-fund-sol-fast } D =$
 (let $(a, D') = \text{square-squarefree-part-nat } D$
 in
 if $D' = 0 \vee D' = 1$ then $(0, 0)$
 else if $a = 1$ then $\text{pell.fund-sol } D$

```

else map-prod id ( $\lambda y. y \text{ div } a$ )
  (shd (sdrop-while ( $\lambda(-, y). y = 0 \vee \neg a \text{ dvd } y$ ) (pell-solutions  $D'$ ))))

```

lemma *find-fund-sol-fast*: $\text{find-fund-sol } D = \text{find-fund-sol-fast } D$
 $\langle \text{proof} \rangle$

end

6 The Connection between the continued fraction expansion of square roots and Pell's equation

theory *Pell-Continued-Fraction*

imports

Sqrt-Nat-Cfrac

Pell.Pell-Algorithm

Polynomial-Factorization.Prime-Factorization

Pell-Lifting

begin

lemma *irrational-times-int-eq-intD*:

assumes $p * \text{real-of-int } a = \text{real-of-int } b$

assumes $p \notin \mathbb{Q}$

shows $a = 0 \wedge b = 0$

$\langle \text{proof} \rangle$

The solutions to Pell's equation for some non-square D are linked to the continued fraction expansion of \sqrt{D} , which we shall show here.

context

fixes $D :: \text{nat}$ **and** $c \ h \ k \ P \ Q \ l$

assumes *nonsquare*: $\neg \text{is-square } D$

defines $c \equiv \text{cfrac-of-real } (\text{sqrt } D)$

defines $h \equiv \text{conv-num } c$ **and** $k \equiv \text{conv-denom } c$

defines $P \equiv \text{fst} \circ \text{sqrt-remainder-surd } D$ **and** $Q \equiv \text{snd} \circ \text{sqrt-remainder-surd } D$

defines $l \equiv \text{sqrt-nat-period-length } D$

begin

interpretation *pell* D

$\langle \text{proof} \rangle$

lemma *cfrac-length-infinite [simp]*: $\text{cfrac-length } c = \infty$

$\langle \text{proof} \rangle$

lemma *conv-num-denom-pell*:

$h \ 0^{\wedge} 2 - D * k \ 0^{\wedge} 2 < 0$

$m > 0 \implies h \ m^{\wedge} 2 - D * k \ m^{\wedge} 2 = (-1)^{\wedge} \text{Suc } m * Q \ m$

$\langle \text{proof} \rangle$

Every non-trivial solution to Pell's equation is a convergent in the expansion

of \sqrt{D} :

theorem *pell-solution-is-conv*:

assumes $x^2 = \text{Suc } (D * y^2)$ **and** $y > 0$

shows $(\text{int } x, \text{int } y) \in \text{range } (\lambda n. (\text{conv-num } c \ n, \text{conv-denom } c \ n))$

<proof>

Let l be the length of the period in the continued fraction expansion of \sqrt{D} and let h_i and k_i be the numerator and denominator of the i -th convergent. Then the non-trivial solutions of Pell's equation are exactly the pairs of the form (h_{lm-1}, k_{lm-1}) for any m such that lm is even.

lemma *nontriv-solution-iff-conv-num-denom*:

nontriv-solution $(x, y) \longleftrightarrow$

$(\exists m > 0. \text{int } x = h \ (l * m - 1) \wedge \text{int } y = k \ (l * m - 1) \wedge \text{even } (l * m))$

<proof>

Consequently, the fundamental solution is (h_n, k_n) where $n = l - 1$ if l is even and $n = 2l - 1$ otherwise:

lemma *fund-sol-conv-num-denom*:

defines $n \equiv \text{if even } l \text{ then } l - 1 \text{ else } 2 * l - 1$

shows $\text{fund-sol} = (\text{nat } (h \ n), \text{nat } (k \ n))$

<proof>

end

The following algorithm computes the fundamental solution (or the dummy result $(0, 0)$ if D is a square) fairly quickly by computing the continued fraction expansion of \sqrt{D} and then computing the fundamental solution as the appropriate convergent.

lemma *find-fund-sol-code* [code]:

find-fund-sol $D =$

$(\text{let } \text{info} = \text{sqrt-cfrac-info-array } D;$

$l = \text{fst } \text{info}$

$\text{in } \text{if } l = 0 \text{ then } (0, 0) \text{ else}$

let

$c = \text{cfrac-sqrt-nth } \text{info};$

$n = \text{if even } l \text{ then } l - 1 \text{ else } 2 * l - 1$

in

$(\text{nat } (\text{conv-num-fun } c \ n), \text{nat } (\text{conv-denom-fun } c \ n))$)

<proof>

lemma *find-nth-solution-square* [simp]: $\text{is-square } D \implies \text{find-nth-solution } D \ n = (0, 0)$

<proof>

lemma *fst-find-fund-sol-eq-0-iff* [simp]: $\text{fst } (\text{find-fund-sol } D) = 0 \longleftrightarrow \text{is-square } D$

<proof>

Arbitrary solutions can now be computed as powers of the fundamental solution.

lemma *find-nth-solution-code* [code]:

```

find-nth-solution D n =
  (let xy = find-fund-sol D
    in if fst xy = 0 then (0, 0) else efficient-pell-power D xy n)
⟨proof⟩

```

lemma *nth-solution-code* [code]:

```

pell.nth-solution D n =
  (let info = sqrt-cfrac-info-array D;
    l = fst info
    in if l = 0 then
      Code.abort (STR "nth-solution is undefined for perfect square parameter.")
        (λ-. pell.nth-solution D n)
    else
      let
        c = cfrac-sqrt-nth info;
        m = if even l then l - 1 else 2 * l - 1;
        fund-sol = (nat (conv-num-fun c m), nat (conv-denom-fun c m))
      in
        efficient-pell-power D fund-sol n)
⟨proof⟩

```

lemma *fund-sol-code* [code]:

```

pell.fund-sol D = (let info = sqrt-cfrac-info-array D;
  l = fst info
  in if l = 0 then
    Code.abort (STR "fund-sol is undefined for perfect square parameter.")
      (λ-. pell.fund-sol D)
  else
    let
      c = cfrac-sqrt-nth info;
      n = if even l then l - 1 else 2 * l - 1
    in
      (nat (conv-num-fun c n), nat (conv-denom-fun c n)))
⟨proof⟩

```

end

7 Tests for Continued Fractions of Square Roots and Pell's Equation

theory *Pell-Continued-Fraction-Tests*

imports

```

Pell.Efficient-Discrete-Sqrt
HOL-Library.Code-Lazy
HOL-Library.Code-Target-Numeral

```

Pell-Continued-Fraction
Pell-Lifting
begin

code-lazy-type *stream*

lemma *lnth-code* [*code*]:

lnth xs 0 = (if lnull xs then undefined (0 :: nat) else lhd xs)
lnth xs (Suc n) = (if lnull xs then undefined (Suc n) else lnth (ltl xs) n)
 ⟨*proof*⟩

value *let c = sqrt-cfrac 1339 in map (cfrac-nth c) [0..<30]*

fun *arg-max-list* **where**

arg-max-list - [] = undefined
 | *arg-max-list f (x # xs) =*
 foldl (λ(x, y) x'. let y' = f x' in if y' > y then (x', y') else (x, y)) (x, f x) xs

value [*code*] *sqrt-cfrac-info 17*

value [*code*] *sqrt-cfrac-info 1339*

value [*code*] *sqrt-cfrac-info 121*

value [*code*] *sqrt-nat-period-length 410286423278424*

For which number $D < 100000$ does \sqrt{D} have the longest period?

value [*code*] *arg-max-list sqrt-nat-period-length [0..<100000]*

7.1 Fundamental solutions of Pell's equation

value [*code*] *pell.fund-sol 12*

value [*code*] *pell.fund-sol 13*

value [*code*] *pell.fund-sol 61*

value [*code*] *pell.fund-sol 661*

value [*code*] *pell.fund-sol 6661*

value [*code*] *pell.fund-sol 4729494*

Project Euler problem #66: For which $D < 1000$ does Pell's equation have the largest fundamental solution?

value [*code*] *arg-max-list (fst ∘ find-fund-sol) [0..<1001]*

The same for $D < 100000$:

value [*code*] *arg-max-list (fst ∘ find-fund-sol) [0..<100000]*

The solution to the next example, which is at the core of Archimedes' cattle problem, is so big that termifying the result takes extremely long. Therefore, we simply compute the number of decimal digits in the result instead.

```

fun log10-aux :: nat ⇒ nat ⇒ nat where
  log10-aux acc n =
    (if n ≥ 10000000000 then log10-aux (acc + 10) (n div 10000000000)
     else if n = 0 then acc else log10-aux (Suc acc) (n div 10))

```

```

definition log10 where log10 = log10-aux 0

```

```

value [code] map-prod log10 log10 (pell.fund-sol 410286423278424)

```

Factoring out the square factor 9314^2 does yield a significant speed-up in this case:

```

value [code] map-prod log10 log10 (find-fund-sol-fast 410286423278424)

```

7.2 Tests for other operations

```

value [code] pell.nth-solution 13 100
value [code] pell.nth-solution 4729494 3

```

```

value [code] stake 10 (pell-solutions 13)
value [code] stake 10 (pell-solutions 61)

```

```

value [code] pell.nth-solution 23 8

```

```

end

```

8 Computing continued fraction expansions through interval arithmetic

```

theory Continued-Fraction-Approximation

```

```

imports

```

```

  Complex-Main
  HOL-Decision-Proc.Approximation
  Coinductive.Coinductive-List
  HOL-Library.Code-Lazy
  HOL-Library.Code-Target-Numeral
  Continued-Fractions

```

```

keywords approximate-cfrac :: diag

```

```

begin

```

The approximation package allows us to compute an enclosing interval for a given real constant. From this, we are able to compute an initial fragment of the continued fraction expansion of the number.

The algorithm essentially works by computing the continued fraction expansion of the lower and upper bound simultaneously and stopping when the results start to diverge.

This algorithm terminates because the lower and upper bounds, being rational numbers, have a finite continued fraction expansion.

definition *float-to-rat* :: *float* ⇒ *int* × *int* **where**

float-to-rat *f* = (if *exponent* *f* ≥ 0 then
(mantissa *f* * 2^{nat} (*exponent* *f*), 1) else (mantissa *f*, 2^{nat} (−*exponent*
f)))

lemma *float-to-rat*: *fst* (*float-to-rat* *f*) / *snd* (*float-to-rat* *f*) = *real-of-float* *f*
⟨*proof*⟩

lemma *snd-float-to-rat-pos* [*simp*]: *snd* (*float-to-rat* *f*) > 0
⟨*proof*⟩

function *cfrac-from-approx* :: *int* × *int* ⇒ *int* × *int* ⇒ *int* list **where**

cfrac-from-approx (*nl*, *dl*) (*nu*, *du*) =
(if *nl* = 0 ∨ *nu* = 0 ∨ *dl* = 0 ∨ *du* = 0 then []
else let *l* = *nl* div *dl*; *u* = *nu* div *du*
in if *l* ≠ *u* then []
else *l* # (let *m* = *nl* mod *dl* in if *m* = 0 then [] else
cfrac-from-approx (*du*, *nu* mod *du*) (*dl*, *m*)))

⟨*proof*⟩

termination ⟨*proof*⟩

lemmas [*simp del*] = *cfrac-from-approx.simps*

lemma *cfrac-from-approx-correct*:

assumes *x* ∈ {*fst* *l* / *snd* *l*..*fst* *u* / *snd* *u*} **and** *snd* *l* > 0 **and** *snd* *u* > 0

assumes *i* < *length* (*cfrac-from-approx* *l* *u*)

shows *cfrac-nth* (*cfrac-of-real* *x*) *i* = *cfrac-from-approx* *l* *u* ! *i*

⟨*proof*⟩

definition *cfrac-from-approx'* :: *float* ⇒ *float* ⇒ *int* list **where**

cfrac-from-approx' *l* *u* = *cfrac-from-approx* (*float-to-rat* *l*) (*float-to-rat* *u*)

lemma *cfrac-from-approx'-correct*:

assumes *x* ∈ {*real-of-float* *l*..*real-of-float* *u*}

assumes *i* < *length* (*cfrac-from-approx'* *l* *u*)

shows *cfrac-nth* (*cfrac-of-real* *x*) *i* = *cfrac-from-approx'* *l* *u* ! *i*

⟨*proof*⟩

definition *approx-cfrac* :: *nat* ⇒ *floatarith* ⇒ *int* list **where**

approx-cfrac *prec* *e* =

(case *approx'* *prec* *e* [] of

None ⇒ []

| Some *ivl* ⇒ *cfrac-from-approx'* (*lower* *ivl*) (*upper* *ivl*))

⟨*ML*⟩

Now let us do some experiments:

value let *prec* = 34; *c* = *cfrac-from-approx'* (*lb-pi* *prec*) (*ub-pi* *prec*) in *c*

```
value let prec = 34; c = cfrac-from-approx' (lb-pi prec) (ub-pi prec)
      in map ( $\lambda n.$  (conv-num-fun (!) c) n, conv-denom-fun (!) c) n) [0.. $\text{length}$ 
c]
```

```
approximate-cfrac prec: 200 pi
approximate-cfrac ln 2
approximate-cfrac exp 1
approximate-cfrac sqrt 129
approximate-cfrac (sqrt 13 + 3) / 4
approximate-cfrac arctan 1
```

```
approximate-cfrac 123 / 97
value cfrac-list-of-rat (123, 97)
```

```
end
```

References

- [1] A. Khinchin and H. Eagle. *Continued Fractions*. Dover books on mathematics. Dover Publications, 1997.
- [2] Proof Wiki.