Farey Sequences and Ford Circles

Lawrence C. Paulson

6 February 2026

Abstract

The sequence F_n of *Farey fractions* of order n has the form

$$\frac{0}{1}, \frac{1}{n}, \frac{1}{n-1}, \ldots, \frac{n-1}{n}, \frac{1}{1}$$

where the fractions appear in numerical order and have denominators at most n. The transformation from F_n to F_{n+1} can be effected by combining adjacent elements of the sequence F_n, using an operation called the *mediant*. Adjacent (reduced) fractions $(a/b) < (c/d)$ satisfy the *unimodular* relation $bc - ad = 1$ and their mediant is $\frac{a+c}{b+d}$. A *Ford circle* is specified by a rational number, and interesting consequences follow in the case of Ford circles obtained from some Farey sequence F_n. The formalised material is drawn from Apostol's *Modular Functions and Dirichlet Series in Number Theory* [1].

1 Farey Sequences and Ford Circles

theory *Farey-Ford*
imports *HOL−Analysis.Analysis HOL−Number-Theory.Totient HOL−Library.Sublist*

begin

lemma *sublist-map-nth*:
assumes $j \leq$ *length xs*
shows *sublist* (*map* ($\lambda i.$ *xs* ! *i*) [*i*..<*j*]) *xs*
⟨*proof*⟩

1.1 Farey sequences

lemma *sorted-two-sublist*:
fixes *x*:: *'a*::*order*
assumes *sorted*: *sorted-wrt* (<) *l*
shows *sublist* [*x*, *y*] *l* ⟷ *x* < *y* ∧ *x* ∈ *set l* ∧ *y* ∈ *set l* ∧ (∀ *z* ∈ *set l*. *z* ≤ *x* ∨ *z* ≥ *y*)
⟨*proof*⟩

lemma *sorted-two-sublist-nth*:
fixes *l*:: *'a*::*order list*
assumes *Suc j* < *length l sorted-wrt* (<) *l*
shows *sublist* [*l* ! *j*, *l* ! *Suc j*] *l*
⟨*proof*⟩

lemma *transp-add1-int*:
assumes ⋀*n*::*int*. *R* (*f n*) (*f* (*1* + *n*))
and *n* < *n'*
and *transp R*
shows *R* (*f n*) (*f n'*)
⟨*proof*⟩

lemma *refl-transp-add1-int*:
assumes ⋀*n*::*int*. *R* (*f n*) (*f* (*1* + *n*))
and *n* ≤ *n'*
and *reflp R transp R*
shows *R* (*f n*) (*f n'*)
⟨*proof*⟩

lemma *transp-Suc*:
assumes ⋀*n*. *R* (*f n*) (*f* (*Suc n*))
and *n* < *n'*
and *transp R*
shows *R* (*f n*) (*f n'*)
⟨*proof*⟩

lemma *refl-transp-Suc*:

assumes $\bigwedge n.\ R\ (f\ n)\ (f\ (Suc\ n))$
and $n \leq n'$
and *reflp R transp R*
shows $R\ (f\ n)\ (f\ n')$
⟨*proof*⟩

lemma *coprime-unimodular-int*:
fixes *a b*::*int*
assumes *coprime a b a*>*1 b*>*1*
obtains *x y* **where** $a*x - b*y = 1\ 0 < x\ x < b\ 0 < y\ y < a$
⟨*proof*⟩

1.2 Farey Fractions

type-synonym *farey* = *rat*

definition *num-farey* :: *farey* ⇒ *int*
where *num-farey* ≡ λ*x*. *fst* (*quotient-of x*)

definition *denom-farey* :: *farey* ⇒ *int*
where *denom-farey* ≡ λ*x*. *snd* (*quotient-of x*)

definition *farey* :: [*int*,*int*] ⇒ *farey*
where *farey* ≡ λ*a b*. *max 0* (*min 1* (*Fract a b*))

lemma *farey01* [*simp*]: $0 \leq farey\ a\ b\ farey\ a\ b \leq 1$
⟨*proof*⟩

lemma *farey-0* [*simp*]: *farey 0 n* = *0*
⟨*proof*⟩

lemma *farey-1* [*simp*]: *farey 1 1* = *1*
⟨*proof*⟩

lemma *num-farey-nonneg*: $x \in \{0..1\} \Longrightarrow num\text{-}farey\ x \geq 0$
⟨*proof*⟩

lemma *num-farey-le-denom*: $x \in \{0..1\} \Longrightarrow num\text{-}farey\ x \leq denom\text{-}farey\ x$
⟨*proof*⟩

lemma *denom-farey-pos*: $denom\text{-}farey\ x > 0$
⟨*proof*⟩

lemma *coprime-num-denom-farey* [*intro*]: *coprime* (*num-farey x*) (*denom-farey x*)
⟨*proof*⟩

lemma *rat-of-farey-conv-num-denom*:
x = *rat-of-int* (*num-farey x*) / *rat-of-int* (*denom-farey x*)
⟨*proof*⟩

lemma *num-denom-farey-eqI*:
assumes $x = \textit{of-int}\ a\ /\ \textit{of-int}\ b\ b > 0\ \textit{coprime}\ a\ b$
shows $\textit{num-farey}\ x = a\ \textit{denom-farey}\ x = b$
⟨*proof*⟩

lemma *farey-cases* [*cases type*, *case-names farey*]:
assumes $x \in \{0..1\}$
obtains $a\ b$ **where** $0 \leq a\ a \leq b\ \textit{coprime}\ a\ b\ x = \textit{Fract}\ a\ b$
⟨*proof*⟩

lemma *rat-of-farey*: $[\![x = \textit{of-int}\ a\ /\ \textit{of-int}\ b;\ x \in \{0..1\}]\!] \Longrightarrow x = \textit{farey}\ a\ b$
⟨*proof*⟩

lemma *farey-num-denom-eq* [*simp*]: $x \in \{0..1\} \Longrightarrow \textit{farey}\ (\textit{num-farey}\ x)\ (\textit{denom-farey}\ x) = x$
⟨*proof*⟩

lemma *farey-eqI*:
assumes $\textit{num-farey}\ x = \textit{num-farey}\ y\ \textit{denom-farey}\ x = \textit{denom-farey}\ y$
shows $x = y$
⟨*proof*⟩

lemma
assumes $\textit{coprime}\ a\ b\ 0 \leq a\ a < b$
shows *num-farey-eq* [*simp*]: $\textit{num-farey}\ (\textit{farey}\ a\ b) = a$
and *denom-farey-eq* [*simp*]: $\textit{denom-farey}\ (\textit{farey}\ a\ b) = b$
⟨*proof*⟩

lemma
assumes $0 \leq a\ a \leq b\ 0 < b$
shows *num-farey*: $\textit{num-farey}\ (\textit{farey}\ a\ b) = a\ \textit{div}\ (\textit{gcd}\ a\ b)$
and *denom-farey*: $\textit{denom-farey}\ (\textit{farey}\ a\ b) = b\ \textit{div}\ (\textit{gcd}\ a\ b)$
⟨*proof*⟩

lemma
assumes $\textit{coprime}\ a\ b\ 0 < b$
shows *num-farey-Fract* [*simp*]: $\textit{num-farey}\ (\textit{Fract}\ a\ b) = a$
and *denom-farey-Fract* [*simp*]: $\textit{denom-farey}\ (\textit{Fract}\ a\ b) = b$
⟨*proof*⟩

lemma *num-farey-0* [*simp*]: $\textit{num-farey}\ 0 = 0$
and *denom-farey-0* [*simp*]: $\textit{denom-farey}\ 0 = 1$
and *num-farey-1* [*simp*]: $\textit{num-farey}\ 1 = 1$
and *denom-farey-1* [*simp*]: $\textit{denom-farey}\ 1 = 1$
⟨*proof*⟩

lemma *num-farey-neq-denom*: $\textit{denom-farey}\ x \neq 1 \Longrightarrow \textit{num-farey}\ x \neq \textit{denom-farey}\ x$

⟨*proof*⟩

lemma *num-farey-0-iff* [*simp*]: *num-farey x = 0 ⟷ x = 0*
⟨*proof*⟩

lemma *denom-farey-le1-cases*:
assumes *denom-farey x ≤ 1 x ∈ {0..1}*
shows *x = 0 ∨ x = 1*
⟨*proof*⟩

definition *mediant* :: *farey ⇒ farey ⇒ farey* **where**
mediant ≡ λx y. Fract (fst (quotient-of x) + fst (quotient-of y))
(snd (quotient-of x) + snd (quotient-of y))

lemma *mediant-eq-Fract*:
mediant x y = Fract (num-farey x + num-farey y) (denom-farey x + denom-farey y)
⟨*proof*⟩

lemma *mediant-eq-farey*:
assumes *x ∈ {0..1} y ∈ {0..1}*
shows *mediant x y = farey (num-farey x + num-farey y) (denom-farey x + denom-farey y)*
⟨*proof*⟩

definition *farey-unimodular* :: *farey ⇒ farey ⇒ bool* **where**
farey-unimodular x y ⟷
denom-farey x ∗ num-farey y − num-farey x ∗ denom-farey y = 1

lemma *farey-unimodular-imp-less*:
assumes *farey-unimodular x y*
shows *x < y*
⟨*proof*⟩

lemma *denom-mediant*: *denom-farey (mediant x y) ≤ denom-farey x + denom-farey y*
⟨*proof*⟩

lemma *unimodular-imp-both-coprime*:
fixes *a*:: *'a*::{*algebraic-semidom*,*comm-ring-1*}
assumes *b∗c − a∗d = 1*
shows *coprime a b coprime c d*
⟨*proof*⟩

lemma *unimodular-imp-coprime*:
fixes *a*:: *'a*::{*algebraic-semidom*,*comm-ring-1*}
assumes *b∗c − a∗d = 1*
shows *coprime (a+c) (b+d)*

⟨*proof*⟩

definition *fareys* :: *nat* ⇒ *rat list*
  where *fareys n* ≡ *sorted-list-of-set* {*x* ∈ {*0..1*}. *denom-farey x* ≤ *n*}

lemma *strict-sorted-fareys*: *sorted-wrt* (<) (*fareys n*) **and** *sorted-fareys*: *sorted* (*fareys n*)
  ⟨*proof*⟩

lemma *distinct-fareys*: *distinct* (*fareys n*)
  ⟨*proof*⟩

lemma *farey-set-UN-farey*: {*x* ∈ {*0..1*}. *denom-farey x* ≤ *n*} = (⋃ *b* ∈ {*1..n*}. ⋃ *a* ∈ {*0..b*}. {*farey a b*})
⟨*proof*⟩

lemma *farey-set-UN-farey'*: {*x* ∈ {*0..1*}. *denom-farey x* ≤ *n*} = (⋃ *b* ∈ {*1..n*}. ⋃ *a* ∈ {*0..b*}. *if coprime a b then* {*farey a b*} *else* {})
⟨*proof*⟩

lemma *farey-set-UN-Fract*: {*x* ∈ {*0..1*}. *denom-farey x* ≤ *n*} = (⋃ *b* ∈ {*1..n*}. ⋃ *a* ∈ {*0..b*}. {*Fract a b*})
  ⟨*proof*⟩

lemma *farey-set-UN-Fract'*: {*x* ∈ {*0..1*}. *denom-farey x* ≤ *n*} = (⋃ *b* ∈ {*1..n*}. ⋃ *a* ∈ {*0..b*}. *if coprime a b then* {*Fract a b*} *else* {})
  ⟨*proof*⟩

lemma *finite-farey-set*: *finite* {*x* ∈ {*0..1*}. *denom-farey x* ≤ *n*}
  ⟨*proof*⟩

lemma *denom-in-fareys-iff*: *x* ∈ *set* (*fareys n*) ⟷ *denom-farey x* ≤ *int n* ∧ *x* ∈ {*0..1*}
  ⟨*proof*⟩

lemma *denom-fareys-leI*: *x* ∈ *set* (*fareys n*) ⟹ *denom-farey x* ≤ *n*
  ⟨*proof*⟩

lemma *denom-fareys-leD*: ⟦*denom-farey x* ≤ *int n*; *x* ∈ {*0..1*}⟧ ⟹ *x* ∈ *set* (*fareys n*)
  ⟨*proof*⟩

lemma *fareys-increasing-1*: *set* (*fareys n*) ⊆ *set* (*fareys* (*Suc n*))
  ⟨*proof*⟩

lemma *fareys-1-minus-half*:
  assumes *r* ∈ *set* (*fareys n*)
  shows *1*−*r* ∈ *set*(*fareys n*)
⟨*proof*⟩

lemma *fareys-1-minus-image*: $(-)1$ ‘ *set* (*fareys* n) = *set* (*fareys* n)
⟨*proof*⟩

lemma *fareys-eq-rev-fareys*: *fareys* n = *rev* (*map* $((-)1)$ (*fareys* n))
⟨*proof*⟩

lemma *fareys-opposite*: $i <$ *length*(*fareys* n) $\Longrightarrow$ *fareys* n ! i = 1 − *fareys* n ! (*length*(*fareys* n) − *Suc* i)
⟨*proof*⟩

lemma *fareys-nonempty*: $n > 0 \Longrightarrow$ *fareys* $n \neq []$
⟨*proof*⟩

lemma *hd-fareys* [*simp*]:
assumes $n > 0$
shows *hd* (*fareys* n) = 0
⟨*proof*⟩

lemma *last-fareys* [*simp*]:
assumes $n > 0$
shows *last* (*fareys* n) = 1
⟨*proof*⟩

1.3 Creating Farey sequences layer by layer

Specifying the precise denominator

definition *fareys-new* :: *nat* ⇒ *rat set* **where**
fareys-new $n \equiv \{Fract\ a\ n |\ a.\ coprime\ a\ n \wedge a \in \{0..n\}\}$

lemma *fareys-new-0* [*simp*]: *fareys-new* $0 = \{\}$
⟨*proof*⟩

lemma *fareys-new-1* [*simp*]: *fareys-new* $1 = \{0,1\}$
⟨*proof*⟩

lemma *fareys-new-not01*:
assumes $n>1$
shows $0 \notin$ (*fareys-new* n) $1 \notin$ (*fareys-new* n)
⟨*proof*⟩

lemma *inj-num-farey*: *inj-on num-farey* (*fareys-new* n)
⟨*proof*⟩

lemma *finite-fareys-new* [*simp*]: *finite* (*fareys-new* n)
⟨*proof*⟩

lemma *card-fareys-new*:
assumes $n > 1$

shows *card* (*fareys-new n*) = *totient n*
⟨*proof*⟩

lemma *disjoint-fareys-plus1*:
assumes *n* > *0*
shows *disjnt* (*set* (*fareys n*)) (*fareys-new* (*Suc n*))
⟨*proof*⟩

lemma *set-fareys-Suc*: *set* (*fareys* (*Suc n*)) = *set* (*fareys n*) ∪ *fareys-new* (*Suc n*)
⟨*proof*⟩

lemma *length-fareys-Suc*:
assumes *n*>*0*
shows *length* (*fareys* (*Suc n*)) = *length* (*fareys n*) + *totient* (*Suc n*)
⟨*proof*⟩

lemma *fareys-0* [*simp*]: *fareys 0* = []
⟨*proof*⟩

lemma *fareys-1* [*simp*]: *fareys 1* = [*0*, *1*]
⟨*proof*⟩

lemma *fareys-2* [*simp*]: *fareys 2* = [*0*, *1*/*2*, *1*]
⟨*proof*⟩

lemma *length-fareys-1*: *length* (*fareys 1*) = *1* + *totient 1*
⟨*proof*⟩

lemma *length-fareys-Suc-if*:
shows *length* (*fareys* (*Suc n*)) = (*if n*=*0 then Suc* (*totient 1*) *else length* (*fareys n*) + *totient* (*Suc n*))
⟨*proof*⟩

lemma *length-fareys*: *n*>*0* ⟹ *length* (*fareys n*) = *1* + (∑ *k*=*1*..*n*. *totient k*)
⟨*proof*⟩

lemma *length-fareys-ge2*:
assumes *n* > *0*
shows *length* (*fareys n*) ≥ *2*
⟨*proof*⟩

lemma *subseq-fareys-1*: *subseq* (*fareys n*) (*fareys* (*Suc n*))
⟨*proof*⟩

lemma *monotone-fareys*: *monotone* (≤) *subseq fareys*
⟨*proof*⟩

lemma *farey-unimodular-0-1* [*simp*, *intro*]: *farey-unimodular 0 1*
⟨*proof*⟩

lemma *fareys-have-01*: $n > 0 \Longrightarrow \{0,1\} \subseteq set\ (fareys\ n)$
⟨*proof*⟩

Apostol's Theorem 5.2 for integers

lemma *mediant-lies-betw-int*:
fixes *a b c d*::*int*
assumes *rat-of-int a / of-int b < of-int c / of-int d b>0 d>0*
shows *rat-of-int a / of-int b < (of-int a + of-int c) / (of-int b + of-int d)*
(rat-of-int a + of-int c) / (of-int b + of-int d) < of-int c / of-int d
⟨*proof*⟩

Apostol's Theorem 5.2

theorem *mediant-inbetween*:
fixes *x y*::*farey*
assumes $x < y$
shows $x < mediant\ x\ y\ mediant\ x\ y < y$
⟨*proof*⟩

lemma *sublist-fareysD*:
assumes *sublist* [*x*,*y*] (*fareys n*)
obtains $x \in set\ (fareys\ n)\ y \in set\ (fareys\ n)$
⟨*proof*⟩

Adding the denominators of two consecutive Farey fractions

lemma *sublist-fareys-add-denoms*:
fixes *a b c d*::*int*
defines $x \equiv Fract\ a\ b$
defines $y \equiv Fract\ c\ d$
assumes *sub*: *sublist* [*x*,*y*] (*fareys n*) **and** *b>0 d>0 coprime a b coprime c d*
shows $b + d > n$
⟨*proof*⟩

1.4 Apostol's Theorems 5.3–5.5

theorem *consec-subset-fareys*:
fixes *a b c d*::*int*
assumes *abcd*: $0 \le Fract\ a\ b\ Fract\ a\ b < Fract\ c\ d\ Fract\ c\ d \le 1$
and *consec*: $b*c - a*d = 1$
and *max*: $max\ b\ d \le n\ n < b+d$
and *b>0*
shows *sublist* [*Fract a b*, *Fract c d*] (*fareys n*)
⟨*proof*⟩

lemma *farey-unimodular-mediant*:
assumes *farey-unimodular x y*
shows *farey-unimodular x* (*mediant x y*) *farey-unimodular* (*mediant x y*) *y*
⟨*proof*⟩

Apostol's Theorem 5.4

theorem *mediant-unimodular*:
fixes *a b c d*::*int*
assumes *abcd*: $0 \leq$ *Fract a b Fract a b* $<$ *Fract c d Fract c d* ≤ 1
and *consec*: $b*c - a*d = 1$
and *0*: $b>0$ $d>0$
defines $h \equiv a+c$
defines $k \equiv b+d$
obtains *Fract a b* $<$ *Fract h k Fract h k* $<$ *Fract c d coprime h k*
$b*h - a*k = 1$ $c*k - d*h = 1$
⟨*proof*⟩

Apostol's Theorem 5.5, first part: "Each fraction in $F\ (n + 1)$ which is not in $F\ n$ is the mediant of a pair of consecutive fractions in $F\ n$

lemma *get-consecutive-parents*:
fixes *m n*::*int*
assumes *coprime m n* $0<m$ $m<n$
obtains *a b c d* **where** $m = a+c$ $n = b+d$ $b*c - a*d = 1$ $a\geq 0$ $b>0$ $c>0$ $d>0$ $a<b$ $c\leq d$
⟨*proof*⟩

theorem *fareys-new-eq-mediant*:
assumes $x \in$ *fareys-new n* $n>1$
obtains *a b c d* **where**
sublist [*Fract a b*, *Fract c d*] (*fareys* $(n-1)$)
$x =$ *mediant* (*Fract a b*) (*Fract c d*) *coprime a b coprime c d* $a\geq 0$ $b>0$ $c>0$ $d>0$
⟨*proof*⟩

Apostol's Theorem 5.5, second part: "Moreover, if $a\ /\ b < c\ /\ d$ are consecutive in any $F\ n$, then they satisfy the unimodular relation $bc - ad = 1$.

theorem *consec-imp-unimodular*:
assumes *sublist* [*Fract a b*, *Fract c d*] (*fareys n*) $b>0$ $d>0$ *coprime a b coprime c d*
shows $b*c - a*d = 1$
⟨*proof*⟩

1.5 Ford circles

definition *Ford-center* :: *rat* ⇒ *complex* **where**
Ford-center r ≡ (λ(*h*,*k*). *Complex* (*h*/*k*) (*1*/(*2* ∗ *k*^*2*))) (*quotient-of r*)

definition *Ford-radius* :: *rat* ⇒ *real* **where**
Ford-radius r ≡ (λ(*h*,*k*). *1*/(*2* ∗ *k*^*2*)) (*quotient-of r*)

definition *Ford-tan* :: [*rat*,*rat*] ⇒ *bool* **where**
Ford-tan r s ≡ *dist* (*Ford-center r*) (*Ford-center s*) = *Ford-radius r* + *Ford-radius s*

definition *Ford-circle* :: *rat* ⇒ *complex set* **where**
Ford-circle r ≡ *sphere* (*Ford-center r*) (*Ford-radius r*)

lemma *Im-Ford-center* [*simp*]: *Im* (*Ford-center r*) = *Ford-radius r*
⟨*proof*⟩

lemma *Ford-radius-nonneg*: *Ford-radius r* ≥ *0*
⟨*proof*⟩

lemma *two-Ford-tangent*:
assumes *r*: (*a*,*b*) = *quotient-of r* **and** *s*: (*c*,*d*) = *quotient-of s*
shows (*dist* (*Ford-center r*) (*Ford-center s*)) ^ *2* − (*Ford-radius r* + *Ford-radius s*) ^ *2*
= ((*a*∗*d* − *b*∗*c*) ^ *2* − *1*) / (*b*∗*d*) ^ *2*
⟨*proof*⟩

Apostol's Theorem 5.6

lemma *two-Ford-tangent-iff*:
assumes *r*: (*a*,*b*) = *quotient-of r* **and** *s*: (*c*,*d*) = *quotient-of s*
shows *Ford-tan r s* ⟷ |*b* ∗ *c* − *a* ∗ *d*| = *1*
⟨*proof*⟩

Also Apostol's Theorem 5.6: Distinct Ford circles do not overlap

lemma *Ford-no-overlap*:
assumes *r* ≠ *s*
shows *dist* (*Ford-center r*) (*Ford-center s*) ≥ *Ford-radius r* + *Ford-radius s*
⟨*proof*⟩

lemma *Ford-aux1*:
assumes *a*≠*0*
shows *cmod* (*Complex* (*b* / (*a* ∗ (a^2 + b^2))) (*1* / (*2* ∗ a^2) − *inverse* (a^2 + b^2)))
= *1* / (*2* ∗ a^2)
(**is** *cmod ?z* = *?r*)
⟨*proof*⟩

lemma *Ford-aux2*:
assumes *a*≠*0*
shows *cmod* (*Complex* (*a* / (*b* ∗ (b^2 + a^2)) − *1* / (*a* ∗ *b*)) (*1* / (*2* ∗ a^2) − *inverse* (b^2 + a^2))) = *1* / (*2* ∗ a^2)
(**is** *cmod ?z* = *?r*)
⟨*proof*⟩

The Rademacher transformation (for theorem 5.8)

definition *Radem-trans* :: *rat* ⇒ *complex* ⇒ *complex* **where**
Radem-trans ≡ λ*r* τ. *let* (*h*,*k*) = *quotient-of r in* − i ∗ *of-int k* ^ *2* ∗ (τ − *of-rat r*)

Theorem 5.8 first part

lemma *Radem-trans-image*: *Radem-trans r* ‘ *Ford-circle r* = *sphere* (*1*/*2*) (*1*/*2*)

⟨*proof*⟩

For the last part of theorem 5.9

lemma *RMS-calc*:
assumes $b + a > int\ N\ N{>}0$
shows $1\ /\ sqrt\ (a^2 + b^2) < sqrt\ 2\ /\ N$
⟨*proof*⟩

locale *three-Ford* =
fixes *N*::*nat*
fixes *h1 k1 h k h2 k2*::*int*
assumes *sub1*: *sublist* [*Fract h1 k1*, *Fract h k*] (*fareys N*)
assumes *sub2*: *sublist* [*Fract h k*, *Fract h2 k2*] (*fareys N*)
assumes *coprime*: *coprime h1 k1 coprime h k coprime h2 k2*
assumes *k-pos*: *k1 > 0 k > 0 k2 > 0*

begin

definition *r1* ≡ *Fract h1 k1*
definition *r* ≡ *Fract h k*
definition *r2* ≡ *Fract h2 k2*

lemma *N-pos*: *N>0*
⟨*proof*⟩

lemma *r-eq-quotient*:
(h1,k1) = quotient-of r (h,k) = quotient-of r (h2,k2) = quotient-of r2
⟨*proof*⟩

lemma *r-eq-divide*:
r1 = of-int h1 / of-int k1 r = of-int h / of-int k r2 = of-int h2/ of-int k2
⟨*proof*⟩

lemma *collapse-r*:
real-of-int h1 / of-int k1 = of-rat r1
real-of-int h / of-int k = of-rat r real-of-int h2/ of-int k2 = of-rat r2
⟨*proof*⟩

lemma *unimod1*: *k1∗h − h1∗k = 1*
and *unimod2*: *k∗h2 − h∗k2 = 1*
⟨*proof*⟩

lemma *r-less*: *r1 < r r < r2*
⟨*proof*⟩

lemma *r01*:
obtains *r1* ∈ {*0..1*} *r* ∈ {*0..1*} *r2* ∈ {*0..1*}
⟨*proof*⟩

lemma *atMost-N*:
obtains $k1 \leq N\ k \leq N\ k2 \leq N$
⟨*proof*⟩

lemma *greaterN1*: $k1 + k > N$
⟨*proof*⟩

lemma *greaterN2*: $k + k2 > N$
⟨*proof*⟩

definition *alpha1* ≡ *Complex* $(h/k - k1$ / *of-int*$(k * (k^2 + k1^2)))$ (*inverse* (*of-int* $(k^2 + k1^2)))$
definition *alpha2* ≡ *Complex* $(h/k + k2$ / *of-int*$(k * (k^2 + k2^2)))$ (*inverse* (*of-int* $(k^2 + k2^2)))$

definition *zed1* ≡ *Complex* (k^2) $(k{*}k1)$ / $((k^2 + k1^2))$
definition *zed2* ≡ *Complex* (k^2) $(-\ k{*}k2)$ / $((k^2 + k2^2))$

Apostol's Theorem 5.7

lemma *three-Ford-tangent*:
obtains *alpha1* ∈ *Ford-circle r alpha1* ∈ *Ford-circle r1*
alpha2 ∈ *Ford-circle r alpha2* ∈ *Ford-circle r2*
⟨*proof*⟩

Theorem 5.8 second part, for alpha1

lemma *Radem-trans-alpha1*: *Radem-trans r alpha1* = *zed1*
⟨*proof*⟩

Theorem 5.8 second part, for alpha2

lemma *Radem-trans-alpha2*: *Radem-trans r alpha2* = *zed2*
⟨*proof*⟩

Theorem 5.9, for zed1

lemma *cmod-zed1*: *cmod zed1* = k / *sqrt* $(k^2 + k1^2)$
⟨*proof*⟩

Theorem 5.9, for zed2

lemma *cmod-zed2*: *cmod zed2* = k / *sqrt* $(k^2 + k2^2)$
⟨*proof*⟩

lemma *on-chord-bounded-cmod*:
assumes *z* ∈ *closed-segment zed1 zed2*
shows *cmod z* < *sqrt 2* ∗ k/N
⟨*proof*⟩

lemma *chord-length-less*: *dist zed1 zed2* < *2* ∗ *sqrt 2* ∗ k / N
⟨*proof*⟩

end

1.6 Material for `Farey_Ford`

The point of tangency between the Ford circles corresponding to the given two rational numbers. It is assumed that x and y are consecutive Farey fractions, in that order.

definition *Ford-tanp* :: *rat* ⇒ *rat* ⇒ *complex* **where**
Ford-tanp x y =
(*let* (*h1*,*k1*) = *quotient-of x*; (*h2*,*k2*) = *quotient-of y*; *m* = *k1^2+k2^2*
in Complex (*h1* / *k1* + *k2* / (*k1* ∗ *m*)) (*1* / *m*))

lemma *Im-Ford-tanp-pos*: *Im* (*Ford-tanp x y*) > *0*
⟨*proof*⟩

lemma *Ford-tanp-on-Ford-circle1*: *Ford-tanp x y* ∈ *Ford-circle x*
⟨*proof*⟩

lemma *Ford-tanp-on-Ford-circle2*:
assumes *farey-unimodular x y*
shows *Ford-tanp x y* ∈ *Ford-circle y*
⟨*proof*⟩

lemma *farey-unimodular-fareys*:
assumes *Suc j* < *length* (*fareys n*)
shows *farey-unimodular* (*fareys n* ! *j*) (*fareys n* ! *Suc j*)
⟨*proof*⟩

lemma *quotient-of-divide* [*simp*]: ⟦*coprime h k*; *k*>*0*⟧ ⟹ *quotient-of* (*rat-of-int h* / *rat-of-int k*) = (*h*,*k*)
⟨*proof*⟩

lemma *quotient-of-divide-nat* [*simp*]: ⟦*coprime h k*; *k*>*0*⟧ ⟹ *quotient-of* (*of-nat h* / *of-nat k*) = (*int h*, *int k*)
⟨*proof*⟩

lemma *quotient-of-oneover-pos-int*: *n* > *0* ⟹ *quotient-of* (*1* / *of-int n*) = (*1*, *n*)
⟨*proof*⟩

lemma *quotient-of-oneover-pos-nat*: *n* > *0* ⟹ *quotient-of* (*1* / *of-nat n*) = (*1*, *int n*)
⟨*proof*⟩

lemma *nth-fareys-1*:
assumes *n* > *0*
shows *fareys n* ! *Suc 0* = *1* / *of-nat n* (**is** - = *?rhs*)
⟨*proof*⟩

lemma *nth-fareys-second-to-last*:

assumes $n > 0$
shows *fareys n* ! (*length* (*fareys n*) − *2*) = *of-nat* (*n* − *1*) / *of-nat n* (**is** *?lhs* = *?rhs*)
⟨*proof*⟩

end

Acknowledgements Manual Eberl set up the initial Farey development.

References

[1] T. M. Apostol. *Modular Functions and Dirichlet Series in Number Theory*. Springer, 1990.