Miscellaneous HOL-Complex Examples

October 1, 2005

Contents

1 Binary arithmetic examples

theory *BinEx*
imports *Complex-Main*
begin

Examples of performing binary arithmetic by simplification. This time we use the reals, though the representation is just of integers.

1.1 Real Arithmetic

1.1.1 Addition

lemma $(1359::real) + -2468 = -1109$
⟨*proof*⟩

lemma $(93746::real) + -46375 = 47371$
⟨*proof*⟩

1.1.2 Negation

lemma $- (65745::real) = -65745$
⟨*proof*⟩

lemma $- (-54321::real) = 54321$
⟨*proof*⟩

1.1.3 Multiplication

lemma $(-84::real) * 51 = -4284$
⟨*proof*⟩

lemma $(255::real) * 255 = 65025$
⟨*proof*⟩

lemma $(1359::real) * -2468 = -3354012$
⟨*proof*⟩

1.1.4 Inequalities

lemma $(89::real) * 10 \neq 889$
⟨*proof*⟩

lemma $(13::real) < 18 - 4$
⟨*proof*⟩

lemma $(-345::real) < -242 + -100$
⟨*proof*⟩

lemma $(13557456::real) < 18678654$

⟨*proof*⟩

lemma (*999999*::*real*) ≤ (*1000001* + *1*) − *2*
⟨*proof*⟩

lemma (*1234567*::*real*) ≤ *1234567*
⟨*proof*⟩

1.1.5 Powers

lemma *2* ^ *15* = (*32768*::*real*)
⟨*proof*⟩

lemma −*3* ^ *7* = (−*2187*::*real*)
⟨*proof*⟩

lemma *13* ^ *7* = (*62748517*::*real*)
⟨*proof*⟩

lemma *3* ^ *15* = (*14348907*::*real*)
⟨*proof*⟩

lemma −*5* ^ *11* = (−*48828125*::*real*)
⟨*proof*⟩

1.1.6 Tests

lemma (*x* + *y* = *x*) = (*y* = (*0*::*real*))
⟨*proof*⟩

lemma (*x* + *y* = *y*) = (*x* = (*0*::*real*))
⟨*proof*⟩

lemma (*x* + *y* = (*0*::*real*)) = (*x* = −*y*)
⟨*proof*⟩

lemma (*x* + *y* = (*0*::*real*)) = (*y* = −*x*)
⟨*proof*⟩

lemma ((*x* + *y*) < (*x* + *z*)) = (*y* < (*z*::*real*))
⟨*proof*⟩

lemma ((*x* + *z*) < (*y* + *z*)) = (*x* < (*y*::*real*))
⟨*proof*⟩

lemma (¬ *x* < *y*) = (*y* ≤ (*x*::*real*))
⟨*proof*⟩

lemma ¬ (*x* < *y* ∧ *y* < (*x*::*real*))
⟨*proof*⟩

lemma $(x::real) < y ==> \neg\ y < x$
⟨*proof*⟩

lemma $((x::real) \neq y) = (x < y \vee y < x)$
⟨*proof*⟩

lemma $(\neg\ x \leq y) = (y < (x::real))$
⟨*proof*⟩

lemma $x \leq y \vee y \leq (x::real)$
⟨*proof*⟩

lemma $x \leq y \vee y < (x::real)$
⟨*proof*⟩

lemma $x < y \vee y \leq (x::real)$
⟨*proof*⟩

lemma $x \leq (x::real)$
⟨*proof*⟩

lemma $((x::real) \leq y) = (x < y \vee x = y)$
⟨*proof*⟩

lemma $((x::real) \leq y \wedge y \leq x) = (x = y)$
⟨*proof*⟩

lemma $\neg(x < y \wedge y \leq (x::real))$
⟨*proof*⟩

lemma $\neg(x \leq y \wedge y < (x::real))$
⟨*proof*⟩

lemma $(-x < (0::real)) = (0 < x)$
⟨*proof*⟩

lemma $((0::real) < -x) = (x < 0)$
⟨*proof*⟩

lemma $(-x \leq (0::real)) = (0 \leq x)$
⟨*proof*⟩

lemma $((0::real) \leq -x) = (x \leq 0)$
⟨*proof*⟩

lemma $(x::real) = y \vee x < y \vee y < x$
⟨*proof*⟩

lemma $(x::real) = 0 \vee 0 < x \vee 0 < -x$
⟨*proof*⟩

lemma $(0::real) \leq x \vee 0 \leq -x$
⟨*proof*⟩

lemma $((x::real) + y \leq x + z) = (y \leq z)$
⟨*proof*⟩

lemma $((x::real) + z \leq y + z) = (x \leq y)$
⟨*proof*⟩

lemma $(w::real) < x \wedge y < z \Longrightarrow w + y < x + z$
⟨*proof*⟩

lemma $(w::real) \leq x \wedge y \leq z \Longrightarrow w + y \leq x + z$
⟨*proof*⟩

lemma $(0::real) \leq x \wedge 0 \leq y \Longrightarrow 0 \leq x + y$
⟨*proof*⟩

lemma $(0::real) < x \wedge 0 < y \Longrightarrow 0 < x + y$
⟨*proof*⟩

lemma $(-x < y) = (0 < x + (y::real))$
⟨*proof*⟩

lemma $(x < -y) = (x + y < (0::real))$
⟨*proof*⟩

lemma $(y < x + -z) = (y + z < (x::real))$
⟨*proof*⟩

lemma $(x + -y < z) = (x < z + (y::real))$
⟨*proof*⟩

lemma $x \leq y \Longrightarrow x < y + (1::real)$
⟨*proof*⟩

lemma $(x - y) + y = (x::real)$
⟨*proof*⟩

lemma $y + (x - y) = (x::real)$
⟨*proof*⟩

lemma $x - x = (0::real)$
⟨*proof*⟩

lemma $(x - y = 0) = (x = (y::real))$

⟨*proof*⟩

lemma $((0::real) \leq x + x) = (0 \leq x)$
⟨*proof*⟩

lemma $(-x \leq x) = ((0::real) \leq x)$
⟨*proof*⟩

lemma $(x \leq -x) = (x \leq (0::real))$
⟨*proof*⟩

lemma $(-x = (0::real)) = (x = 0)$
⟨*proof*⟩

lemma $-(x - y) = y - (x::real)$
⟨*proof*⟩

lemma $((0::real) < x - y) = (y < x)$
⟨*proof*⟩

lemma $((0::real) \leq x - y) = (y \leq x)$
⟨*proof*⟩

lemma $(x + y) - x = (y::real)$
⟨*proof*⟩

lemma $(-x = y) = (x = (-y::real))$
⟨*proof*⟩

lemma $x < (y::real) ==> \neg(x = y)$
⟨*proof*⟩

lemma $(x \leq x + y) = ((0::real) \leq y)$
⟨*proof*⟩

lemma $(y \leq x + y) = ((0::real) \leq x)$
⟨*proof*⟩

lemma $(x < x + y) = ((0::real) < y)$
⟨*proof*⟩

lemma $(y < x + y) = ((0::real) < x)$
⟨*proof*⟩

lemma $(x - y) - x = (-y::real)$
⟨*proof*⟩

lemma $(x + y < z) = (x < z - (y::real))$
⟨*proof*⟩

lemma $(x - y < z) = (x < z + (y::real))$
⟨*proof*⟩

lemma $(x < y - z) = (x + z < (y::real))$
⟨*proof*⟩

lemma $(x \leq y - z) = (x + z \leq (y::real))$
⟨*proof*⟩

lemma $(x - y \leq z) = (x \leq z + (y::real))$
⟨*proof*⟩

lemma $(-x < -y) = (y < (x::real))$
⟨*proof*⟩

lemma $(-x \leq -y) = (y \leq (x::real))$
⟨*proof*⟩

lemma $(a + b) - (c + d) = (a - c) + (b - (d::real))$
⟨*proof*⟩

lemma $(0::real) - x = -x$
⟨*proof*⟩

lemma $x - (0::real) = x$
⟨*proof*⟩

lemma $w \leq x \wedge y < z ==> w + y < x + (z::real)$
⟨*proof*⟩

lemma $w < x \wedge y \leq z ==> w + y < x + (z::real)$
⟨*proof*⟩

lemma $(0::real) \leq x \wedge 0 < y ==> 0 < x + (y::real)$
⟨*proof*⟩

lemma $(0::real) < x \wedge 0 \leq y ==> 0 < x + y$
⟨*proof*⟩

lemma $-x - y = -(x + (y::real))$
⟨*proof*⟩

lemma $x - (-y) = x + (y::real)$
⟨*proof*⟩

lemma $-x - -y = y - (x::real)$
⟨*proof*⟩

lemma $(a - b) + (b - c) = a - (c::real)$
⟨*proof*⟩

lemma $(x = y - z) = (x + z = (y::real))$
⟨*proof*⟩

lemma $(x - y = z) = (x = z + (y::real))$
⟨*proof*⟩

lemma $x - (x - y) = (y::real)$
⟨*proof*⟩

lemma $x - (x + y) = -(y::real)$
⟨*proof*⟩

lemma $x = y ==> x \leq (y::real)$
⟨*proof*⟩

lemma $(0::real) < x ==> \neg(x = 0)$
⟨*proof*⟩

lemma $(x + y) * (x - y) = (x * x) - (y * y)$
⟨*proof*⟩

lemma $(-x = -y) = (x = (y::real))$
⟨*proof*⟩

lemma $(-x < -y) = (y < (x::real))$
⟨*proof*⟩

lemma $!!a::real.\ a \leq b ==> c \leq d ==> x + y < z ==> a + c \leq b + d$
⟨*proof*⟩

lemma $!!a::real.\ a < b ==> c < d ==> a - d \leq b + (-c)$
⟨*proof*⟩

lemma $!!a::real.\ a \leq b ==> b + b \leq c ==> a + a \leq c$
⟨*proof*⟩

lemma $!!a::real.\ a + b \leq i + j ==> a \leq b ==> i \leq j ==> a + a \leq j + j$
⟨*proof*⟩

lemma $!!a::real.\ a + b < i + j ==> a < b ==> i < j ==> a + a < j + j$
⟨*proof*⟩

lemma $!!a::real.\ a + b + c \leq i + j + k \wedge a \leq b \wedge b \leq c \wedge i \leq j \wedge j \leq k --> a + a + a \leq k + k + k$
⟨*proof*⟩

lemma !!a::*real*. $a + b + c + d \leq i + j + k + l$ ==> $a \leq b$ ==> $b \leq c$
==> $c \leq d$ ==> $i \leq j$ ==> $j \leq k$ ==> $k \leq l$ ==> $a \leq l$
⟨*proof*⟩

lemma !!a::*real*. $a + b + c + d \leq i + j + k + l$ ==> $a \leq b$ ==> $b \leq c$
==> $c \leq d$ ==> $i \leq j$ ==> $j \leq k$ ==> $k \leq l$ ==> $a + a + a + a \leq l + l + l + l$
⟨*proof*⟩

lemma !!a::*real*. $a + b + c + d \leq i + j + k + l$ ==> $a \leq b$ ==> $b \leq c$
==> $c \leq d$ ==> $i \leq j$ ==> $j \leq k$ ==> $k \leq l$ ==> $a + a + a + a + a \leq l + l + l + l + i$
⟨*proof*⟩

lemma !!a::*real*. $a + b + c + d \leq i + j + k + l$ ==> $a \leq b$ ==> $b \leq c$
==> $c \leq d$ ==> $i \leq j$ ==> $j \leq k$ ==> $k \leq l$ ==> $a + a + a + a + a + a \leq l + l + l + l + i + l$
⟨*proof*⟩

1.2 Complex Arithmetic

lemma $(1359 + 93746*ii) - (2468 + 46375*ii) = -1109 + 47371*ii$
⟨*proof*⟩

lemma $-(65745 + -47371*ii) = -65745 + 47371*ii$
⟨*proof*⟩

Multiplication requires distributive laws. Perhaps versions instantiated to literal constants should be added to the simpset.

lemmas *distrib* = *left-distrib right-distrib left-diff-distrib right-diff-distrib*

lemma $(1 + ii) * (1 - ii) = 2$
⟨*proof*⟩

lemma $(1 + 2*ii) * (1 + 3*ii) = -5 + 5*ii$
⟨*proof*⟩

lemma $(-84 + 255*ii) + (51 * 255*ii) = -84 + 13260 * ii$
⟨*proof*⟩

No inequalities or linear arithmetic: the complex numbers are unordered!

No powers (not supported yet)

end

2 Square roots of primes are irrational

theory *Sqrt*

imports *Primes Complex-Main*
begin

2.1 Preliminaries

The set of rational numbers, including the key representation theorem.

constdefs
  rationals :: *real set* ($\mathbb{Q}$)
  $\mathbb{Q} \equiv \{x.\ \exists\, m\ n.\ n \neq 0 \wedge |x| = real\ (m::nat)\ /\ real\ (n::nat)\}$

theorem *rationals-rep*: $x \in \mathbb{Q} \Longrightarrow$
  $\exists\, m\ n.\ n \neq 0 \wedge |x| = real\ m\ /\ real\ n \wedge gcd\ (m,\ n) = 1$
⟨*proof*⟩

lemma [*elim?*]: $r \in \mathbb{Q} \Longrightarrow$
  $(\bigwedge m\ n.\ n \neq 0 \Longrightarrow |r| = real\ m\ /\ real\ n \Longrightarrow gcd\ (m,\ n) = 1 \Longrightarrow C)$
    $\Longrightarrow C$
  ⟨*proof*⟩

2.2 Main theorem

The square root of any prime number (including *2*) is irrational.

theorem *sqrt-prime-irrational*: *prime p* $\Longrightarrow$ *sqrt* (*real p*) $\notin \mathbb{Q}$
⟨*proof*⟩

corollary *sqrt* (*real* (*2*::*nat*)) $\notin \mathbb{Q}$
  ⟨*proof*⟩

2.3 Variations

Here is an alternative version of the main proof, using mostly linear forward-reasoning. While this results in less top-down structure, it is probably closer to proofs seen in mathematics.

theorem *prime p* $\Longrightarrow$ *sqrt* (*real p*) $\notin \mathbb{Q}$
⟨*proof*⟩

end

3 Square roots of primes are irrational (script version)

theory *Sqrt-Script*
imports *Primes Complex-Main*
begin

Contrast this linear Isabelle/Isar script with Markus Wenzel's more mathematical version.

3.1 Preliminaries

lemma *prime-nonzero*: $prime\ p \Longrightarrow p \neq 0$
⟨*proof*⟩

lemma *prime-dvd-other-side*:
$n * n = p * (k * k) \Longrightarrow prime\ p \Longrightarrow p\ dvd\ n$
⟨*proof*⟩

lemma *reduction*: $prime\ p \Longrightarrow$
$0 < k \Longrightarrow k * k = p * (j * j) \Longrightarrow k < p * j \wedge 0 < j$
⟨*proof*⟩

lemma *rearrange*: $(j::nat) * (p * j) = k * k \Longrightarrow k * k = p * (j * j)$
⟨*proof*⟩

lemma *prime-not-square*:
$prime\ p \Longrightarrow (\bigwedge k.\ 0 < k \Longrightarrow m * m \neq p * (k * k))$
⟨*proof*⟩

3.2 The set of rational numbers

constdefs
$rationals :: real\ set \quad (\mathbb{Q})$
$\mathbb{Q} \equiv \{x.\ \exists m\ n.\ n \neq 0 \wedge |x| = real\ (m::nat)\ /\ real\ (n::nat)\}$

3.3 Main theorem

The square root of any prime number (including *2*) is irrational.

theorem *prime-sqrt-irrational*:
$prime\ p \Longrightarrow x * x = real\ p \Longrightarrow 0 \leq x \Longrightarrow x \notin \mathbb{Q}$
⟨*proof*⟩

lemmas *two-sqrt-irrational* =
prime-sqrt-irrational [*OF two-is-prime*]

end

4 The Nonstandard Primes as an Extension of the Prime Numbers

theory *NSPrimes*
imports ~~/*src/HOL/NumberTheory/Factorization Complex-Main*

begin

These can be used to derive an alternative proof of the infinitude of primes by considering a property of nonstandard sets.

constdefs
hdvd :: [*hypnat*, *hypnat*] => *bool* (**infixl** *hdvd 50*)
(*M*::*hypnat*) *hdvd N* == (∗*p2*∗ (*op dvd*)) *M N*

declare *hdvd-def* [*transfer-unfold*]

constdefs
starprime :: *hypnat set*
starprime == (∗*s*∗ {*p*. *prime p*})

declare *starprime-def* [*transfer-unfold*]

constdefs
choicefun :: *'a set* => *'a*
choicefun E == (@*x*. ∃ *X* ∈ *Pow*(*E*) −{{}}. *x* : *X*)

consts *injf-max* :: *nat* => (*'a*::{*order*} *set*) => *'a*
primrec
injf-max-zero: *injf-max 0 E* = *choicefun E*
injf-max-Suc: *injf-max* (*Suc n*) *E* = *choicefun*({*e*. *e*:*E* & *injf-max n E* < *e*})

A "choice" theorem for ultrafilters, like almost everywhere quantification

lemma *UF-choice*: {*n*. ∃ *m*. *Q n m*} : *FreeUltrafilterNat*
==> ∃ *f*. {*n*. *Q n* (*f n*)} : *FreeUltrafilterNat*
⟨*proof*⟩

lemma *UF-if*: ({*n*. *P n*} : *FreeUltrafilterNat* −−> {*n*. *Q n*} : *FreeUltrafilterNat*)
=
({*n*. *P n* −−> *Q n*} : *FreeUltrafilterNat*)
⟨*proof*⟩

lemma *UF-conj*: ({*n*. *P n*} : *FreeUltrafilterNat* & {*n*. *Q n*} : *FreeUltrafilterNat*)
=
({*n*. *P n* & *Q n*} : *FreeUltrafilterNat*)
⟨*proof*⟩

lemma *UF-choice-ccontr*: (∀ *f*. {*n*. *Q n* (*f n*)} : *FreeUltrafilterNat*) =
({*n*. ∀ *m*. *Q n m*} : *FreeUltrafilterNat*)
⟨*proof*⟩

lemma *dvd-by-all*: ∀ *M*. ∃ *N*. 0 < *N* & (∀ *m*. 0 < *m* & (*m*::*nat*) <= *M* −−> *m dvd N*)
⟨*proof*⟩

lemmas *dvd-by-all2* = *dvd-by-all* [*THEN spec*, *standard*]

lemma *lemma-hypnat-P-EX*: $(\exists (x::hypnat).\ P\ x) = (\exists f.\ P\ (star\text{-}n\ f))$
⟨*proof*⟩

lemma *lemma-hypnat-P-ALL*: $(\forall (x::hypnat).\ P\ x) = (\forall f.\ P\ (star\text{-}n\ f))$
⟨*proof*⟩

lemma *hdvd*:
$(star\text{-}n\ X\ hdvd\ star\text{-}n\ Y) =$
$(\{n.\ X\ n\ dvd\ Y\ n\} : FreeUltrafilterNat)$
⟨*proof*⟩

lemma *hypnat-of-nat-le-zero-iff*: $(hypnat\text{-}of\text{-}nat\ n <= 0) = (n = 0)$
⟨*proof*⟩
declare *hypnat-of-nat-le-zero-iff* [*simp*]

lemma *hdvd-by-all*: $\forall M.\ \exists N.\ 0 < N\ \&\ (\forall m.\ 0 < m\ \&\ (m::hypnat) <= M\ --> m\ hdvd\ N)$
⟨*proof*⟩

lemmas *hdvd-by-all2* = *hdvd-by-all* [*THEN spec*, *standard*]

lemma *hypnat-dvd-all-hypnat-of-nat*:
$\exists (N::hypnat).\ 0 < N\ \&\ (\forall n \in -\{0::nat\}.\ hypnat\text{-}of\text{-}nat(n)\ hdvd\ N)$
⟨*proof*⟩

The nonstandard extension of the set prime numbers consists of precisely those hypernaturals exceeding 1 that have no nontrivial factors

lemma *starprime*:
$starprime = \{p.\ 1 < p\ \&\ (\forall m.\ m\ hdvd\ p\ --> m = 1 \mid m = p)\}$
⟨*proof*⟩

lemma *prime-two*: *prime 2*
⟨*proof*⟩
declare *prime-two* [*simp*]

lemma *prime-factor-exists* [*rule-format*]: $Suc\ 0 < n\ --> (\exists k.\ prime\ k\ \&\ k\ dvd\ n)$
⟨*proof*⟩

lemma *hyperprime-factor-exists* [*rule-format*]:
$!!n.\ 1 < n ==> (\exists k \in starprime.\ k\ hdvd\ n)$
⟨*proof*⟩

lemma *NatStar-hypnat-of-nat*: *finite A ==> ∗s∗ A = hypnat-of-nat ‘ A*
⟨*proof*⟩

lemma *FreeUltrafilterNat-singleton-not-mem*: $\{x\} \notin$ *FreeUltrafilterNat*
⟨*proof*⟩
declare *FreeUltrafilterNat-singleton-not-mem* [*simp*]

4.1 Another characterization of infinite set of natural numbers

lemma *finite-nat-set-bounded*: *finite N ==> ∃ n. (∀ i ∈ N. i<(n::nat))*
⟨*proof*⟩

lemma *finite-nat-set-bounded-iff*: *finite N = (∃ n. (∀ i ∈ N. i<(n::nat)))*
⟨*proof*⟩

lemma *not-finite-nat-set-iff*: *(~ finite N) = (∀ n. ∃ i ∈ N. n <= (i::nat))*
⟨*proof*⟩

lemma *bounded-nat-set-is-finite2*: *(∀ i ∈ N. i<=(n::nat)) ==> finite N*
⟨*proof*⟩

lemma *finite-nat-set-bounded2*: *finite N ==> ∃ n. (∀ i ∈ N. i<=(n::nat))*
⟨*proof*⟩

lemma *finite-nat-set-bounded-iff2*: *finite N = (∃ n. (∀ i ∈ N. i<=(n::nat)))*
⟨*proof*⟩

lemma *not-finite-nat-set-iff2*: *(~ finite N) = (∀ n. ∃ i ∈ N. n < (i::nat))*
⟨*proof*⟩

4.2 An injective function cannot define an embedded natural number

lemma *lemma-infinite-set-singleton*: *∀ m n. m ≠ n --> f n ≠ f m*
==> {n. f n = N} = {} | (∃ m. {n. f n = N} = {m})
⟨*proof*⟩

lemma *inj-fun-not-hypnat-in-SHNat*: *inj (f::nat=>nat) ==> star-n f ∉ Nats*
⟨*proof*⟩

lemma *range-subset-mem-starsetNat*:
range f <= A ==> star-n f ∈ ∗s∗ A
⟨*proof*⟩

lemma *lemmaPow3*: $E \neq \{\} \Longrightarrow \exists x.\ \exists X \in (Pow\ E - \{\{\}\}).\ x: X$
⟨*proof*⟩

lemma *choicefun-mem-set*: $E \neq \{\} \Longrightarrow choicefun\ E \in E$
⟨*proof*⟩
declare *choicefun-mem-set* [*simp*]

lemma *injf-max-mem-set*: $[|\ E \neq\{\};\ \forall x.\ \exists y \in E.\ x < y\ |] \Longrightarrow injf\text{-}max\ n\ E \in E$
⟨*proof*⟩

lemma *injf-max-order-preserving*: $\forall x.\ \exists y \in E.\ x < y \Longrightarrow injf\text{-}max\ n\ E < injf\text{-}max\ (Suc\ n)\ E$
⟨*proof*⟩

lemma *injf-max-order-preserving2*: $\forall x.\ \exists y \in E.\ x < y$
$\Longrightarrow \forall n\ m.\ m < n \longrightarrow injf\text{-}max\ m\ E < injf\text{-}max\ n\ E$
⟨*proof*⟩

lemma *inj-injf-max*: $\forall x.\ \exists y \in E.\ x < y \Longrightarrow inj\ (\%n.\ injf\text{-}max\ n\ E)$
⟨*proof*⟩

lemma *infinite-set-has-order-preserving-inj*:
$[|\ (E::('a::\{order\}\ set)) \neq \{\};\ \forall x.\ \exists y \in E.\ x < y\ |]$
$\Longrightarrow \exists f.\ range\ f <= E\ \&\ inj\ (f::nat \Rightarrow 'a)\ \&\ (\forall m.\ f\ m < f(Suc\ m))$
⟨*proof*⟩

Only need the existence of an injective function from N to A for proof

lemma *hypnat-infinite-has-nonstandard*:
$\sim$ *finite A* $\Longrightarrow$ *hypnat-of-nat* ‘ *A* $<$ ($*s*$ *A*)
⟨*proof*⟩

lemma *starsetNat-eq-hypnat-of-nat-image-finite*: $*s*\ A = $ *hypnat-of-nat* ‘ *A* $\Longrightarrow$ *finite A*
⟨*proof*⟩

lemma *finite-starsetNat-iff*: ($*s*$ *A* = *hypnat-of-nat* ‘ *A*) = (*finite A*)
⟨*proof*⟩

lemma *hypnat-infinite-has-nonstandard-iff*: ($\sim$ *finite A*) = (*hypnat-of-nat* ‘ *A* $<$ $*s*$ *A*)
⟨*proof*⟩

4.3 Existence of Infinitely Many Primes: a Nonstandard Proof

lemma *lemma-not-dvd-hypnat-one*: $\sim (\forall n \in -\{0\}.\ hypnat\text{-}of\text{-}nat\ n\ hdvd\ 1)$
⟨*proof*⟩
declare *lemma-not-dvd-hypnat-one* [*simp*]

lemma *lemma-not-dvd-hypnat-one2*: $\exists n \in -\{0\}.\ \sim hypnat\text{-}of\text{-}nat\ n\ hdvd\ 1$
⟨*proof*⟩
declare *lemma-not-dvd-hypnat-one2* [*simp*]

lemma *hypnat-gt-zero-gt-one*:
$!!N.\ [|\ 0 < (N::hypnat);\ N \neq 1\ |] ==> 1 < N$
⟨*proof*⟩

lemma *hypnat-add-one-gt-one*:
$!!N.\ 0 < N ==> 1 < (N::hypnat) + 1$
⟨*proof*⟩

lemma *zero-not-prime*: $\neg\ prime\ 0$
⟨*proof*⟩
declare *zero-not-prime* [*simp*]

lemma *hypnat-of-nat-zero-not-prime*: $hypnat\text{-}of\text{-}nat\ 0 \notin starprime$
⟨*proof*⟩
declare *hypnat-of-nat-zero-not-prime* [*simp*]

lemma *hypnat-zero-not-prime*:
$0 \notin starprime$
⟨*proof*⟩
declare *hypnat-zero-not-prime* [*simp*]

lemma *one-not-prime*: $\neg\ prime\ 1$
⟨*proof*⟩
declare *one-not-prime* [*simp*]

lemma *one-not-prime2*: $\neg\ prime(Suc\ 0)$
⟨*proof*⟩
declare *one-not-prime2* [*simp*]

lemma *hypnat-of-nat-one-not-prime*: $hypnat\text{-}of\text{-}nat\ 1 \notin starprime$
⟨*proof*⟩
declare *hypnat-of-nat-one-not-prime* [*simp*]

lemma *hypnat-one-not-prime*: $1 \notin starprime$
⟨*proof*⟩
declare *hypnat-one-not-prime* [*simp*]

lemma *hdvd-diff*: $!!k\ m\ n.\ [|\ k\ hdvd\ m;\ k\ hdvd\ n\ |] ==> k\ hdvd\ (m - n)$

⟨*proof*⟩

lemma *dvd-one-eq-one*: *x dvd* (*1*::*nat*) ==> *x* = *1*
⟨*proof*⟩

lemma *hdvd-one-eq-one*: !!*x*. *x hdvd 1* ==> *x* = *1*
⟨*proof*⟩

theorem *not-finite-prime*: ~ *finite* {*p*. *prime p*}
⟨*proof*⟩

end

5 Big O notation – continued

theory *BigO-Complex*
imports *BigO Complex*
begin

Additional lemmas that require the `HOL-Complex` logic image.

lemma *bigo-LIMSEQ1*: *f* =*o* *O*(*g*) ==> *g* ----> *0* ==> *f* ----> *0*
⟨*proof*⟩

lemma *bigo-LIMSEQ2*: *f* =*o* *g* +*o* *O*(*h*) ==> *h* ----> *0* ==> *f* ----> *a*
==> *g* ----> *a*
⟨*proof*⟩

end