

# Isabelle/HOL-NSA — Non-Standard Analysis

May 23, 2024

## Contents

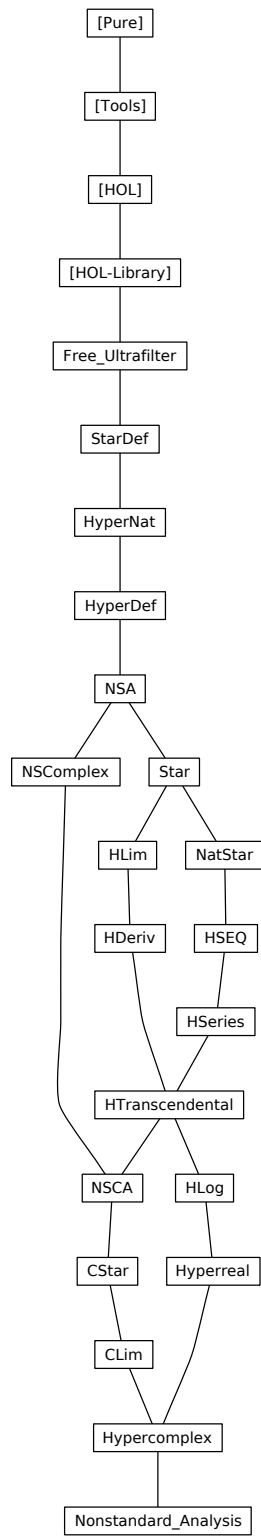
<b>1</b>	<b>Filters and Ultrafilters</b>	<b>3</b>
1.1	Definitions and basic properties . . . . .	3
1.1.1	Ultrafilters . . . . .	3
1.2	Maximal filter = Ultrafilter . . . . .	3
1.3	Ultrafilter Theorem . . . . .	4
1.3.1	Free Ultrafilters . . . . .	4
<b>2</b>	<b>Construction of Star Types Using Ultrafilters</b>	<b>5</b>
2.1	A Free Ultrafilter over the Naturals . . . . .	5
2.2	Definition of <i>star</i> type constructor . . . . .	5
2.3	Transfer principle . . . . .	6
2.4	Standard elements . . . . .	8
2.5	Internal functions . . . . .	8
2.6	Internal predicates . . . . .	9
2.7	Internal sets . . . . .	10
2.8	Syntactic classes . . . . .	12
2.9	Ordering and lattice classes . . . . .	16
2.10	Ordered group classes . . . . .	17
2.11	Ring and field classes . . . . .	18
2.12	Power . . . . .	20
2.13	Number classes . . . . .	21
2.14	Finite class . . . . .	22
<b>3</b>	<b>Hypernatural numbers</b>	<b>22</b>
3.1	Properties Transferred from Naturals . . . . .	22
3.2	Properties of the set of embedded natural numbers . . . . .	24
3.3	Infinite Hypernatural Numbers – <i>HNatInfinite</i> . . . . .	25
3.3.1	Closure Rules . . . . .	25
3.4	Existence of an infinite hypernatural number . . . . .	26
3.4.1	Alternative characterization of the set of infinite hypernaturals . . . . .	27

3.4.2	Alternative Characterization of <i>HNatInfinite</i> using Free Ultrafilter . . . . .	27
3.5	Embedding of the Hypernaturals into other types . . . . .	28
<b>4</b>	<b>Construction of Hyperreals Using Ultrafilters</b>	<b>29</b>
4.1	Real vector class instances . . . . .	29
4.2	Injection from <i>hypreal</i> . . . . .	30
4.3	Properties of <i>starrel</i> . . . . .	31
4.4	<i>hypreal-of-real</i> : the Injection from <i>real</i> to <i>hypreal</i> . . . . .	31
4.5	Properties of <i>star-n</i> . . . . .	32
4.6	Existence of Infinite Hyperreal Number . . . . .	32
4.7	Embedding the Naturals into the Hyperreals . . . . .	33
4.8	Exponentials on the Hyperreals . . . . .	33
4.9	Powers with Hypernatural Exponents . . . . .	34
<b>5</b>	<b>Infinite Numbers, Infinitesimals, Infinitely Close Relation</b>	<b>36</b>
5.1	Nonstandard Extension of the Norm Function . . . . .	37
5.2	Closure Laws for the Standard Reals . . . . .	39
5.3	Set of Finite Elements is a Subring of the Extended Reals . . . . .	40
5.4	Set of Infinitesimals is a Subring of the Hyperreals . . . . .	41
5.5	The Infinitely Close Relation . . . . .	45
5.6	Zero is the Only Infinitesimal that is also a Real . . . . .	49
<b>6</b>	<b>Standard Part Theorem</b>	<b>51</b>
6.1	Uniqueness: Two Infinitely Close Reals are Equal . . . . .	51
6.2	Existence of Unique Real Infinitely Close . . . . .	52
6.2.1	Lifting of the Ub and Lub Properties . . . . .	52
6.3	Finite, Infinite and Infinitesimal . . . . .	53
6.4	Theorems about Monads . . . . .	56
6.5	Proof that $x \approx y$ implies $ x  \approx  y $ . . . . .	57
6.6	More <i>HFinite</i> and <i>Infinitesimal</i> Theorems . . . . .	58
6.7	Theorems about Standard Part . . . . .	59
6.8	Alternative Definitions using Free Ultrafilter . . . . .	61
6.8.1	<i>HFinite</i> . . . . .	61
6.8.2	<i>HInfinite</i> . . . . .	62
6.8.3	<i>Infinitesimal</i> . . . . .	62
6.9	Proof that $\omega$ is an infinite number . . . . .	63
<b>7</b>	<b>Nonstandard Complex Numbers</b>	<b>65</b>
7.0.1	Real and Imaginary parts . . . . .	65
7.0.2	Imaginary unit . . . . .	65
7.0.3	Complex conjugate . . . . .	65
7.0.4	Argand . . . . .	65
7.0.5	Injection from hyperreals . . . . .	66

7.0.6	$e^{\wedge}(x + iy)$ . . . . .	66
7.1	Properties of Nonstandard Real and Imaginary Parts . . . . .	67
7.2	Addition for Nonstandard Complex Numbers . . . . .	67
7.3	More Minus Laws . . . . .	67
7.4	More Multiplication Laws . . . . .	68
7.5	Subtraction and Division . . . . .	68
7.6	Embedding Properties for <i>hcomplex-of-hypreal</i> Map . . . . .	68
7.7	<i>HComplex</i> theorems . . . . .	68
7.8	Modulus (Absolute Value) of Nonstandard Complex Number	69
7.9	Conjugation . . . . .	70
7.10	More Theorems about the Function <i>hcmmod</i> . . . . .	71
7.11	Exponentiation . . . . .	71
7.12	The Function <i>hsgn</i> . . . . .	72
7.12.1	<i>harg</i> . . . . .	73
7.13	Polar Form for Nonstandard Complex Numbers . . . . .	73
7.14	<i>hcomplex-of-complex</i> : the Injection from type <i>complex</i> to to <i>hcomplex</i> . . . . .	76
7.15	Numerals and Arithmetic . . . . .	76
<b>8</b>	<b>Star-Transforms in Non-Standard Analysis</b>	<b>77</b>
8.1	Preamble - Pulling $\exists$ over $\forall$ . . . . .	77
8.2	Properties of the Star-transform Applied to Sets of Reals . . . . .	78
8.3	Theorems about nonstandard extensions of functions . . . . .	78
<b>9</b>	<b>Star-transforms for the Hypernaturals</b>	<b>82</b>
9.1	Nonstandard Extensions of Functions . . . . .	83
9.2	Nonstandard Characterization of Induction . . . . .	85
<b>10</b>	<b>Sequences and Convergence (Nonstandard)</b>	<b>86</b>
10.1	Limits of Sequences . . . . .	86
10.1.1	Equivalence of <i>LIMSEQ</i> and <i>NSLIMSEQ</i> . . . . .	88
10.1.2	Derived theorems about <i>NSLIMSEQ</i> . . . . .	88
10.2	Convergence . . . . .	89
10.3	Bounded Monotonic Sequences . . . . .	89
10.3.1	Upper Bounds and Lubs of Bounded Sequences . . . . .	90
10.3.2	A Bounded and Monotonic Sequence Converges . . . . .	90
10.4	Cauchy Sequences . . . . .	91
10.4.1	Equivalence Between NS and Standard . . . . .	91
10.4.2	Cauchy Sequences are Bounded . . . . .	91
10.4.3	Cauchy Sequences are Convergent . . . . .	91
10.5	Power Sequences . . . . .	92

<b>11 Finite Summation and Infinite Series for Hyperreals</b>	<b>92</b>
11.1 Nonstandard Sums . . . . .	94
11.2 Infinite sums: Standard and NS theorems . . . . .	95
<b>12 Limits and Continuity (Nonstandard)</b>	<b>96</b>
12.1 Limits of Functions . . . . .	96
12.1.1 Equivalence of <i>filterlim</i> and <i>NSLIM</i> . . . . .	98
12.2 Continuity . . . . .	98
12.3 Uniform Continuity . . . . .	99
<b>13 Differentiation (Nonstandard)</b>	<b>99</b>
13.1 Derivatives . . . . .	100
13.2 Lemmas . . . . .	102
13.2.1 Equivalence of NS and Standard definitions . . . . .	103
13.2.2 Differentiability predicate . . . . .	104
13.3 (NS) Increment . . . . .	104
<b>14 Nonstandard Extensions of Transcendental Functions</b>	<b>105</b>
14.1 Nonstandard Extension of Square Root Function . . . . .	105
14.2 Proving $\sin^*(1/n) \times 1/(1/n) \approx 1$ for $n = \infty$ . . . . .	111
<b>15 Non-Standard Complex Analysis</b>	<b>112</b>
15.1 Closure Laws for SComplex, the Standard Complex Numbers	113
15.2 The Finite Elements form a Subring . . . . .	114
15.3 The Complex Infinitesimals form a Subring . . . . .	114
15.4 The “Infinitely Close” Relation . . . . .	114
15.5 Zero is the Only Infinitesimal Complex Number . . . . .	115
15.6 Properties of <i>hRe</i> , <i>hIm</i> and <i>HComplex</i> . . . . .	116
15.7 Theorems About Monads . . . . .	117
15.8 Theorems About Standard Part . . . . .	117
<b>16 Star-transforms in NSA, Extending Sets of Complex Numbers and Complex Functions</b>	<b>120</b>
16.1 Properties of the *-Transform Applied to Sets of Reals . . . . .	120
16.2 Theorems about Nonstandard Extensions of Functions . . . . .	120
16.3 Internal Functions - Some Redundancy With <i>*f*</i> Now . . . . .	120
<b>17 Limits, Continuity and Differentiation for Complex Functions</b>	<b>120</b>
17.1 Limit of Complex to Complex Function . . . . .	121
17.2 Continuity . . . . .	122
17.3 Functions from Complex to Reals . . . . .	122
17.4 Differentiation of Natural Number Powers . . . . .	122
17.5 Derivative of Reciprocals (Function <i>inverse</i> ) . . . . .	123
17.6 Derivative of Quotient . . . . .	123

17.7 Caratheodory Formulation of Derivative at a Point: Standard Proof . . . . .	123
<b>18 Logarithms: Non-Standard Version</b>	<b>124</b>



## 1 Filters and Ultrafilters

```
theory Free-Ultrafilter
  imports HOL-Library.Infinite-Set
begin
```

### 1.1 Definitions and basic properties

#### 1.1.1 Ultrafilters

```
locale ultrafilter =
  fixes F :: 'a filter
  assumes proper: F ≠ bot
  assumes ultra: eventually P F ∨ eventually (λx. ¬ P x) F
begin
```

```
lemma eventually-imp-frequently: frequently P F ⇒ eventually P F
  ⟨proof⟩
```

```
lemma frequently-eq-eventually: frequently P F = eventually P F
  ⟨proof⟩
```

```
lemma eventually-disj-iff: eventually (λx. P x ∨ Q x) F ↔ eventually P F ∨
  eventually Q F
  ⟨proof⟩
```

```
lemma eventually-all-iff: eventually (λx. ∀ y. P x y) F = (∀ Y. eventually (λx. P
  x (Y x)) F)
  ⟨proof⟩
```

```
lemma eventually-imp-iff: eventually (λx. P x → Q x) F ↔ (eventually P F
  → eventually Q F)
  ⟨proof⟩
```

```
lemma eventually-iff-iff: eventually (λx. P x ↔ Q x) F ↔ (eventually P F
  ↔ eventually Q F)
  ⟨proof⟩
```

```
lemma eventually-not-iff: eventually (λx. ¬ P x) F ↔ ¬ eventually P F
  ⟨proof⟩
```

```
end
```

### 1.2 Maximal filter = Ultrafilter

A filter  $F$  is an ultrafilter iff it is a maximal filter, i.e. whenever  $G$  is a filter and  $F \subseteq G$  then  $F = G$

Lemma that shows existence of an extension to what was assumed to be a maximal filter. Will be used to derive contradiction in proof of property of

ultrafilter.

**lemma** *extend-filter*:  $frequently\ P\ F \implies inf\ F\ (principal\ \{x.\ P\ x\}) \neq bot$   
 ⟨*proof*⟩

**lemma** *max-filter-ultrafilter*:

**assumes**  $F \neq bot$

**assumes** *max*:  $\bigwedge G.\ G \neq bot \implies G \leq F \implies F = G$

**shows** *ultrafilter*  $F$

⟨*proof*⟩

**lemma** *le-filter-frequently*:  $F \leq G \iff (\forall P.\ frequently\ P\ F \implies frequently\ P\ G)$   
 ⟨*proof*⟩

**lemma** (*in ultrafilter*) *max-filter*:

**assumes**  $G: G \neq bot$

**and** *sub*:  $G \leq F$

**shows**  $F = G$

⟨*proof*⟩

### 1.3 Ultrafilter Theorem

**lemma** *ex-max-ultrafilter*:

**fixes**  $F :: 'a\ filter$

**assumes**  $F: F \neq bot$

**shows**  $\exists U \leq F.\ ultrafilter\ U$

⟨*proof*⟩

#### 1.3.1 Free Ultrafilters

There exists a free ultrafilter on any infinite set.

**locale** *freeultrafilter* = *ultrafilter* +

**assumes** *infinite*:  $eventually\ P\ F \implies infinite\ \{x.\ P\ x\}$

**begin**

**lemma** *finite*:  $finite\ \{x.\ P\ x\} \implies \neg eventually\ P\ F$   
 ⟨*proof*⟩

**lemma** *finite'*:  $finite\ \{x.\ \neg P\ x\} \implies eventually\ P\ F$   
 ⟨*proof*⟩

**lemma** *le-cofinite*:  $F \leq cofinite$   
 ⟨*proof*⟩

**lemma** *singleton*:  $\neg eventually\ (\lambda x.\ x = a)\ F$   
 ⟨*proof*⟩

**lemma** *singleton'*:  $\neg eventually\ ((=)\ a)\ F$   
 ⟨*proof*⟩



**lemma** *ultrafilter*: *ultrafilter*  $F$   $\langle$ *proof* $\rangle$

**end**

**lemma** *freeultrafilter-Ex*:

**assumes** [*simp*]: *infinite* ( $UNIV :: 'a$  *set*)

**shows**  $\exists U :: 'a$  *filter*. *freeultrafilter*  $U$

$\langle$ *proof* $\rangle$

**end**

## 2 Construction of Star Types Using Ultrafilters

**theory** *StarDef*

**imports** *Free-Ultrafilter*

**begin**

### 2.1 A Free Ultrafilter over the Naturals

**definition** *FreeUltrafilterNat* :: *nat filter* ( $\langle \mathcal{U} \rangle$ )

**where**  $\mathcal{U} = (\text{SOME } U. \text{freeultrafilter } U)$

**lemma** *freeultrafilter-FreeUltrafilterNat*: *freeultrafilter*  $\mathcal{U}$

$\langle$ *proof* $\rangle$

**interpretation** *FreeUltrafilterNat*: *freeultrafilter*  $\mathcal{U}$

$\langle$ *proof* $\rangle$

### 2.2 Definition of *star* type constructor

**definition** *starrel* ::  $((\text{nat} \Rightarrow 'a) \times (\text{nat} \Rightarrow 'a))$  *set*

**where**  $\text{starrel} = \{(X, Y). \text{eventually } (\lambda n. X\ n = Y\ n)\ \mathcal{U}\}$

**definition** *star* =  $(UNIV :: (\text{nat} \Rightarrow 'a)$  *set*) // *starrel*

**typedef**  $'a$  *star* = *star* ::  $(\text{nat} \Rightarrow 'a)$  *set set*

$\langle$ *proof* $\rangle$

**definition** *star-n* ::  $(\text{nat} \Rightarrow 'a) \Rightarrow 'a$  *star*

**where**  $\text{star-n } X = \text{Abs-star } (\text{starrel } \{\{X\}\})$

**theorem** *star-cases* [*case-names star-n*, *cases type: star*]:

**obtains**  $X$  **where**  $x = \text{star-n } X$

$\langle$ *proof* $\rangle$

**lemma** *all-star-eq*:  $(\forall x. P\ x) \longleftrightarrow (\forall X. P\ (\text{star-n } X))$

$\langle$ *proof* $\rangle$

**lemma** *ex-star-eq*:  $(\exists x. P x) \longleftrightarrow (\exists X. P (\text{star-n } X))$   
 ⟨proof⟩

Proving that *starrel* is an equivalence relation.

**lemma** *starrel-iff* [*iff*]:  $(X, Y) \in \text{starrel} \longleftrightarrow \text{eventually } (\lambda n. X n = Y n) \mathcal{U}$   
 ⟨proof⟩

**lemma** *equiv-starrel*: *equiv UNIV starrel*  
 ⟨proof⟩

**lemmas** *equiv-starrel-iff = eq-equiv-class-iff* [*OF equiv-starrel UNIV-I UNIV-I*]

**lemma** *starrel-in-star*:  $\text{starrel} \{x\} \in \text{star}$   
 ⟨proof⟩

**lemma** *star-n-eq-iff*:  $\text{star-n } X = \text{star-n } Y \longleftrightarrow \text{eventually } (\lambda n. X n = Y n) \mathcal{U}$   
 ⟨proof⟩

### 2.3 Transfer principle

This introduction rule starts each transfer proof.

**lemma** *transfer-start*:  $P \equiv \text{eventually } (\lambda n. Q) \mathcal{U} \Longrightarrow \text{Trueprop } P \equiv \text{Trueprop } Q$   
 ⟨proof⟩

Standard principles that play a central role in the transfer tactic.

**definition** *Ifun* ::  $('a \Rightarrow 'b) \text{ star} \Rightarrow 'a \text{ star} \Rightarrow 'b \text{ star}$  ( $\langle(- \star / -)\rangle$  [300, 301] 300)  
**where** *Ifun* *f*  $\equiv$   
 $\lambda x. \text{Abs-star } (\bigcup F \in \text{Rep-star } f. \bigcup X \in \text{Rep-star } x. \text{starrel} \{ \lambda n. F n (X n) \})$

**lemma** *Ifun-congruent2*: *congruent2 starrel starrel*  $(\lambda F X. \text{starrel} \{ \lambda n. F n (X n) \})$   
 ⟨proof⟩

**lemma** *Ifun-star-n*:  $\text{star-n } F \star \text{star-n } X = \text{star-n } (\lambda n. F n (X n))$   
 ⟨proof⟩

**lemma** *transfer-Ifun*:  $f \equiv \text{star-n } F \Longrightarrow x \equiv \text{star-n } X \Longrightarrow f \star x \equiv \text{star-n } (\lambda n. F n (X n))$   
 ⟨proof⟩

**definition** *star-of* ::  $'a \Rightarrow 'a \text{ star}$   
**where** *star-of* *x*  $\equiv \text{star-n } (\lambda n. x)$

Initialize transfer tactic.

⟨ML⟩

Transfer introduction rules.

**lemma** *transfer-ex* [*transfer-intro*]:

$$\begin{aligned}
& (\bigwedge X. p \text{ (star-} n \text{ } X) \equiv \text{eventually } (\lambda n. P \ n \ (X \ n)) \ \mathcal{U}) \implies \\
& \quad \exists x::'a \text{ star. } p \ x \equiv \text{eventually } (\lambda n. \exists x. P \ n \ x) \ \mathcal{U} \\
& \langle \text{proof} \rangle
\end{aligned}$$

**lemma** *transfer-all* [*transfer-intro*]:

$$\begin{aligned}
& (\bigwedge X. p \text{ (star-} n \text{ } X) \equiv \text{eventually } (\lambda n. P \ n \ (X \ n)) \ \mathcal{U}) \implies \\
& \quad \forall x::'a \text{ star. } p \ x \equiv \text{eventually } (\lambda n. \forall x. P \ n \ x) \ \mathcal{U} \\
& \langle \text{proof} \rangle
\end{aligned}$$

**lemma** *transfer-not* [*transfer-intro*]:  $p \equiv \text{eventually } P \ \mathcal{U} \implies \neg p \equiv \text{eventually } (\lambda n. \neg P \ n) \ \mathcal{U}$   
 $\langle \text{proof} \rangle$

**lemma** *transfer-conj* [*transfer-intro*]:

$$\begin{aligned}
& p \equiv \text{eventually } P \ \mathcal{U} \implies q \equiv \text{eventually } Q \ \mathcal{U} \implies p \wedge q \equiv \text{eventually } (\lambda n. P \ n \wedge \\
& Q \ n) \ \mathcal{U} \\
& \langle \text{proof} \rangle
\end{aligned}$$

**lemma** *transfer-disj* [*transfer-intro*]:

$$\begin{aligned}
& p \equiv \text{eventually } P \ \mathcal{U} \implies q \equiv \text{eventually } Q \ \mathcal{U} \implies p \vee q \equiv \text{eventually } (\lambda n. P \ n \vee \\
& Q \ n) \ \mathcal{U} \\
& \langle \text{proof} \rangle
\end{aligned}$$

**lemma** *transfer-imp* [*transfer-intro*]:

$$\begin{aligned}
& p \equiv \text{eventually } P \ \mathcal{U} \implies q \equiv \text{eventually } Q \ \mathcal{U} \implies p \longrightarrow q \equiv \text{eventually } (\lambda n. P \ n \\
& \longrightarrow Q \ n) \ \mathcal{U} \\
& \langle \text{proof} \rangle
\end{aligned}$$

**lemma** *transfer-iff* [*transfer-intro*]:

$$\begin{aligned}
& p \equiv \text{eventually } P \ \mathcal{U} \implies q \equiv \text{eventually } Q \ \mathcal{U} \implies p = q \equiv \text{eventually } (\lambda n. P \ n = \\
& Q \ n) \ \mathcal{U} \\
& \langle \text{proof} \rangle
\end{aligned}$$

**lemma** *transfer-if-bool* [*transfer-intro*]:

$$\begin{aligned}
& p \equiv \text{eventually } P \ \mathcal{U} \implies x \equiv \text{eventually } X \ \mathcal{U} \implies y \equiv \text{eventually } Y \ \mathcal{U} \implies \\
& \quad (\text{if } p \text{ then } x \text{ else } y) \equiv \text{eventually } (\lambda n. \text{if } P \ n \text{ then } X \ n \text{ else } Y \ n) \ \mathcal{U} \\
& \langle \text{proof} \rangle
\end{aligned}$$

**lemma** *transfer-eq* [*transfer-intro*]:

$$\begin{aligned}
& x \equiv \text{star-} n \ X \implies y \equiv \text{star-} n \ Y \implies x = y \equiv \text{eventually } (\lambda n. X \ n = Y \ n) \ \mathcal{U} \\
& \langle \text{proof} \rangle
\end{aligned}$$

**lemma** *transfer-if* [*transfer-intro*]:

$$\begin{aligned}
& p \equiv \text{eventually } (\lambda n. P \ n) \ \mathcal{U} \implies x \equiv \text{star-} n \ X \implies y \equiv \text{star-} n \ Y \implies \\
& \quad (\text{if } p \text{ then } x \text{ else } y) \equiv \text{star-} n \ (\lambda n. \text{if } P \ n \text{ then } X \ n \text{ else } Y \ n) \\
& \langle \text{proof} \rangle
\end{aligned}$$

**lemma** *transfer-fun-eq* [*transfer-intro*]:

$$\begin{aligned}
& (\bigwedge X. f \text{ (star-} n \ X) = g \text{ (star-} n \ X) \equiv \text{eventually } (\lambda n. F \ n \ (X \ n) = G \ n \ (X \ n))
\end{aligned}$$

$\mathcal{U}) \implies$   
 $f = g \equiv \text{eventually } (\lambda n. F n = G n) \mathcal{U}$   
 ⟨proof⟩

**lemma** *transfer-star-n* [*transfer-intro*]:  $\text{star-n } X \equiv \text{star-n } (\lambda n. X n)$   
 ⟨proof⟩

**lemma** *transfer-bool* [*transfer-intro*]:  $p \equiv \text{eventually } (\lambda n. p) \mathcal{U}$   
 ⟨proof⟩

## 2.4 Standard elements

**definition** *Standard* :: 'a star set  
 where *Standard* = range star-of

Transfer tactic should remove occurrences of *star-of*.

⟨ML⟩

**lemma** *star-of-inject*:  $\text{star-of } x = \text{star-of } y \longleftrightarrow x = y$   
 ⟨proof⟩

**lemma** *Standard-star-of* [*simp*]:  $\text{star-of } x \in \text{Standard}$   
 ⟨proof⟩

## 2.5 Internal functions

Transfer tactic should remove occurrences of *Ifun*.

⟨ML⟩

**lemma** *Ifun-star-of* [*simp*]:  $\text{star-of } f \star \text{star-of } x = \text{star-of } (f x)$   
 ⟨proof⟩

**lemma** *Standard-Ifun* [*simp*]:  $f \in \text{Standard} \implies x \in \text{Standard} \implies f \star x \in \text{Standard}$   
 ⟨proof⟩

Nonstandard extensions of functions.

**definition** *starfun* :: ('a  $\Rightarrow$  'b)  $\Rightarrow$  'a star  $\Rightarrow$  'b star (⟨\*f\* -> [80] 80)  
 where *starfun* f  $\equiv \lambda x. \text{star-of } f \star x$

**definition** *starfun2* :: ('a  $\Rightarrow$  'b  $\Rightarrow$  'c)  $\Rightarrow$  'a star  $\Rightarrow$  'b star  $\Rightarrow$  'c star (⟨\*f2\* -> [80] 80)  
 where *starfun2* f  $\equiv \lambda x y. \text{star-of } f \star x \star y$

**declare** *starfun-def* [*transfer-unfold*]  
**declare** *starfun2-def* [*transfer-unfold*]

**lemma** *starfun-star-n*:  $(\text{*f* } f) (\text{star-n } X) = \text{star-n } (\lambda n. f (X n))$   
 ⟨proof⟩

**lemma** *starfun2-star-n*:  $( *f2* f ) (star-n X) (star-n Y) = star-n (\lambda n. f (X n) (Y n))$   
 ⟨proof⟩

**lemma** *starfun-star-of [simp]*:  $( *f* f ) (star-of x) = star-of (f x)$   
 ⟨proof⟩

**lemma** *starfun2-star-of [simp]*:  $( *f2* f ) (star-of x) = *f* f x$   
 ⟨proof⟩

**lemma** *Standard-starfun [simp]*:  $x \in Standard \implies starfun f x \in Standard$   
 ⟨proof⟩

**lemma** *Standard-starfun2 [simp]*:  $x \in Standard \implies y \in Standard \implies starfun2 f x y \in Standard$   
 ⟨proof⟩

**lemma** *Standard-starfun-iff*:  
**assumes** *inj*:  $\bigwedge x y. f x = f y \implies x = y$   
**shows**  $starfun f x \in Standard \longleftrightarrow x \in Standard$   
 ⟨proof⟩

**lemma** *Standard-starfun2-iff*:  
**assumes** *inj*:  $\bigwedge a b a' b'. f a b = f a' b' \implies a = a' \wedge b = b'$   
**shows**  $starfun2 f x y \in Standard \longleftrightarrow x \in Standard \wedge y \in Standard$   
 ⟨proof⟩

## 2.6 Internal predicates

**definition** *unstar* ::  $bool \Rightarrow bool$   
**where**  $unstar b \longleftrightarrow b = star-of True$

**lemma** *unstar-star-n*:  $unstar (star-n P) \longleftrightarrow eventually P \mathcal{U}$   
 ⟨proof⟩

**lemma** *unstar-star-of [simp]*:  $unstar (star-of p) = p$   
 ⟨proof⟩

Transfer tactic should remove occurrences of *unstar*.

⟨ML⟩

**lemma** *transfer-unstar [transfer-intro]*:  $p \equiv star-n P \implies unstar p \equiv eventually P \mathcal{U}$   
 ⟨proof⟩

**definition** *starP* ::  $('a \Rightarrow bool) \Rightarrow 'a \Rightarrow bool$  ( $\langle *p* \rightarrow [80] 80$ )  
**where**  $*p* P = (\lambda x. unstar (star-of P \star x))$

**definition**  $starP2 :: ('a \Rightarrow 'b \Rightarrow bool) \Rightarrow 'a \text{ star} \Rightarrow 'b \text{ star} \Rightarrow bool$  ( $\langle *p2* \rightarrow [80]$   
80)

**where**  $*p2* P = (\lambda x y. unstar (star\text{-of } P \star x \star y))$

**declare**  $starP\text{-def}$  [*transfer-unfold*]

**declare**  $starP2\text{-def}$  [*transfer-unfold*]

**lemma**  $starP\text{-star-n}: (*p* P) (star\text{-n } X) = eventually (\lambda n. P (X n)) \mathcal{U}$   
 $\langle proof \rangle$

**lemma**  $starP2\text{-star-n}: (*p2* P) (star\text{-n } X) (star\text{-n } Y) = (eventually (\lambda n. P (X n) (Y n))) \mathcal{U}$   
 $\langle proof \rangle$

**lemma**  $starP\text{-star-of}$  [*simp*]:  $(*p* P) (star\text{-of } x) = P x$   
 $\langle proof \rangle$

**lemma**  $starP2\text{-star-of}$  [*simp*]:  $(*p2* P) (star\text{-of } x) = *p* P x$   
 $\langle proof \rangle$

## 2.7 Internal sets

**definition**  $Iset :: 'a \text{ set star} \Rightarrow 'a \text{ star set}$   
**where**  $Iset A = \{x. (*p2* (\in)) x A\}$

**lemma**  $Iset\text{-star-n}: (star\text{-n } X \in Iset (star\text{-n } A)) = (eventually (\lambda n. X n \in A n)) \mathcal{U}$   
 $\langle proof \rangle$

Transfer tactic should remove occurrences of  $Iset$ .

$\langle ML \rangle$

**lemma**  $transfer\text{-mem}$  [*transfer-intro*]:  
 $x \equiv star\text{-n } X \Longrightarrow a \equiv Iset (star\text{-n } A) \Longrightarrow x \in a \equiv eventually (\lambda n. X n \in A n) \mathcal{U}$   
 $\langle proof \rangle$

**lemma**  $transfer\text{-Collect}$  [*transfer-intro*]:  
 $(\bigwedge X. p (star\text{-n } X) \equiv eventually (\lambda n. P n (X n)) \mathcal{U}) \Longrightarrow$   
 $Collect p \equiv Iset (star\text{-n } (\lambda n. Collect (P n)))$   
 $\langle proof \rangle$

**lemma**  $transfer\text{-set-eq}$  [*transfer-intro*]:  
 $a \equiv Iset (star\text{-n } A) \Longrightarrow b \equiv Iset (star\text{-n } B) \Longrightarrow a = b \equiv eventually (\lambda n. A n = B n) \mathcal{U}$   
 $\langle proof \rangle$

**lemma**  $transfer\text{-ball}$  [*transfer-intro*]:  
 $a \equiv Iset (star\text{-n } A) \Longrightarrow (\bigwedge X. p (star\text{-n } X) \equiv eventually (\lambda n. P n (X n)) \mathcal{U}) \Longrightarrow$

$\forall x \in a. p\ x \equiv \text{eventually } (\lambda n. \forall x \in A\ n. P\ n\ x)\ \mathcal{U}$   
 ⟨proof⟩

**lemma** *transfer-beq* [*transfer-intro*]:

$a \equiv \text{Iset } (\text{star-}n\ A) \implies (\bigwedge X. p\ (\text{star-}n\ X) \equiv \text{eventually } (\lambda n. P\ n\ (X\ n))\ \mathcal{U}) \implies$   
 $\exists x \in a. p\ x \equiv \text{eventually } (\lambda n. \exists x \in A\ n. P\ n\ x)\ \mathcal{U}$   
 ⟨proof⟩

**lemma** *transfer-Iset* [*transfer-intro*]:  $a \equiv \text{star-}n\ A \implies \text{Iset } a \equiv \text{Iset } (\text{star-}n\ (\lambda n.$

$A\ n))$   
 ⟨proof⟩

Nonstandard extensions of sets.

**definition** *starset* :: ‘*a* set  $\Rightarrow$  ‘*a* star set ( $\text{star} \rightarrow [80] 80$ )  
 where *starset*  $A = \text{Iset } (\text{star-of } A)$

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

**lemma** *starset-mem*:  $\text{star-of } x \in \text{star} A \iff x \in A$   
 ⟨proof⟩

**lemma** *starset-UNIV*:  $\text{star} (\text{UNIV}::'a\ \text{set}) = (\text{UNIV}::'a\ \text{star set})$   
 ⟨proof⟩

**lemma** *starset-empty*:  $\text{star} \{\} = \{\}$   
 ⟨proof⟩

**lemma** *starset-insert*:  $\text{star} (\text{insert } x\ A) = \text{insert } (\text{star-of } x)\ (\text{star } A)$   
 ⟨proof⟩

**lemma** *starset-Un*:  $\text{star} (A \cup B) = \text{star } A \cup \text{star } B$   
 ⟨proof⟩

**lemma** *starset-Int*:  $\text{star} (A \cap B) = \text{star } A \cap \text{star } B$   
 ⟨proof⟩

**lemma** *starset-Compl*:  $\text{star } -A = -(\text{star } A)$   
 ⟨proof⟩

**lemma** *starset-diff*:  $\text{star} (A - B) = \text{star } A - \text{star } B$   
 ⟨proof⟩

**lemma** *starset-image*:  $\text{star} (f\ `A) = (\text{star } f)\ `(\text{star } A)$   
 ⟨proof⟩

**lemma** *starset-vimage*:  $\text{star} (f\ -`A) = (\text{star } f)\ -`(\text{star } A)$   
 ⟨proof⟩

**lemma** *starset-subset*:  $(\text{star } A \subseteq \text{star } B) \iff A \subseteq B$

*<proof>*

**lemma** *starset-eq*: ( $*s* A = *s* B$ )  $\leftrightarrow A = B$   
*<proof>*

**lemmas** *starset-simps* [*simp*] =  
*starset-mem starset-UNIV*  
*starset-empty starset-insert*  
*starset-Un starset-Int*  
*starset-Compl starset-diff*  
*starset-image starset-vimage*  
*starset-subset starset-eq*

## 2.8 Syntactic classes

**instantiation** *star* :: (*zero*) *zero*  
**begin**  
  **definition** *star-zero-def*:  $0 \equiv \text{star-of } 0$   
  **instance** *<proof>*  
**end**

**instantiation** *star* :: (*one*) *one*  
**begin**  
  **definition** *star-one-def*:  $1 \equiv \text{star-of } 1$   
  **instance** *<proof>*  
**end**

**instantiation** *star* :: (*plus*) *plus*  
**begin**  
  **definition** *star-add-def*:  $(+) \equiv *f2* (+)$   
  **instance** *<proof>*  
**end**

**instantiation** *star* :: (*times*) *times*  
**begin**  
  **definition** *star-mult-def*:  $((*)) \equiv *f2* ((*))$   
  **instance** *<proof>*  
**end**

**instantiation** *star* :: (*uminus*) *uminus*  
**begin**  
  **definition** *star-minus-def*: *uminus*  $\equiv *f* \text{uminus}$   
  **instance** *<proof>*  
**end**

**instantiation** *star* :: (*minus*) *minus*  
**begin**  
  **definition** *star-diff-def*:  $(-) \equiv *f2* (-)$   
  **instance** *<proof>*



**end**

**instantiation** *star* :: (*abs*) *abs*

**begin**

**definition** *star-abs-def*:  $abs \equiv *f* \ abs$

**instance**  $\langle proof \rangle$

**end**

**instantiation** *star* :: (*sgn*) *sgn*

**begin**

**definition** *star-sgn-def*:  $sgn \equiv *f* \ sgn$

**instance**  $\langle proof \rangle$

**end**

**instantiation** *star* :: (*divide*) *divide*

**begin**

**definition** *star-divide-def*:  $divide \equiv *f2* \ divide$

**instance**  $\langle proof \rangle$

**end**

**instantiation** *star* :: (*inverse*) *inverse*

**begin**

**definition** *star-inverse-def*:  $inverse \equiv *f* \ inverse$

**instance**  $\langle proof \rangle$

**end**

**instance** *star* :: (*Rings.dvd*) *Rings.dvd*  $\langle proof \rangle$

**instantiation** *star* :: (*modulo*) *modulo*

**begin**

**definition** *star-mod-def*:  $(mod) \equiv *f2* \ (mod)$

**instance**  $\langle proof \rangle$

**end**

**instantiation** *star* :: (*ord*) *ord*

**begin**

**definition** *star-le-def*:  $(\leq) \equiv *p2* \ (\leq)$

**definition** *star-less-def*:  $(<) \equiv *p2* \ (<)$

**instance**  $\langle proof \rangle$

**end**

**lemmas** *star-class-defs* [*transfer-unfold*] =

*star-zero-def*    *star-one-def*

*star-add-def*    *star-diff-def*    *star-minus-def*

*star-mult-def*    *star-divide-def*    *star-inverse-def*

*star-le-def*    *star-less-def*    *star-abs-def*    *star-sgn-def*

*star-mod-def*

Class operations preserve standard elements.

**lemma** *Standard-zero*:  $0 \in \text{Standard}$   
 ⟨proof⟩

**lemma** *Standard-one*:  $1 \in \text{Standard}$   
 ⟨proof⟩

**lemma** *Standard-add*:  $x \in \text{Standard} \implies y \in \text{Standard} \implies x + y \in \text{Standard}$   
 ⟨proof⟩

**lemma** *Standard-diff*:  $x \in \text{Standard} \implies y \in \text{Standard} \implies x - y \in \text{Standard}$   
 ⟨proof⟩

**lemma** *Standard-minus*:  $x \in \text{Standard} \implies -x \in \text{Standard}$   
 ⟨proof⟩

**lemma** *Standard-mult*:  $x \in \text{Standard} \implies y \in \text{Standard} \implies x * y \in \text{Standard}$   
 ⟨proof⟩

**lemma** *Standard-divide*:  $x \in \text{Standard} \implies y \in \text{Standard} \implies x / y \in \text{Standard}$   
 ⟨proof⟩

**lemma** *Standard-inverse*:  $x \in \text{Standard} \implies \text{inverse } x \in \text{Standard}$   
 ⟨proof⟩

**lemma** *Standard-abs*:  $x \in \text{Standard} \implies |x| \in \text{Standard}$   
 ⟨proof⟩

**lemma** *Standard-mod*:  $x \in \text{Standard} \implies y \in \text{Standard} \implies x \text{ mod } y \in \text{Standard}$   
 ⟨proof⟩

**lemmas** *Standard-simps* [simp] =  
*Standard-zero Standard-one*  
*Standard-add Standard-diff Standard-minus*  
*Standard-mult Standard-divide Standard-inverse*  
*Standard-abs Standard-mod*

*star-of* preserves class operations.

**lemma** *star-of-add*:  $\text{star-of } (x + y) = \text{star-of } x + \text{star-of } y$   
 ⟨proof⟩

**lemma** *star-of-diff*:  $\text{star-of } (x - y) = \text{star-of } x - \text{star-of } y$   
 ⟨proof⟩

**lemma** *star-of-minus*:  $\text{star-of } (-x) = - \text{star-of } x$   
 ⟨proof⟩

**lemma** *star-of-mult*:  $\text{star-of } (x * y) = \text{star-of } x * \text{star-of } y$   
 ⟨proof⟩

**lemma** *star-of-divide*:  $\text{star-of } (x / y) = \text{star-of } x / \text{star-of } y$   
 ⟨proof⟩

**lemma** *star-of-inverse*:  $\text{star-of } (\text{inverse } x) = \text{inverse } (\text{star-of } x)$   
 ⟨proof⟩

**lemma** *star-of-mod*:  $\text{star-of } (x \text{ mod } y) = \text{star-of } x \text{ mod } \text{star-of } y$   
 ⟨proof⟩

**lemma** *star-of-abs*:  $\text{star-of } |x| = |\text{star-of } x|$   
 ⟨proof⟩

*star-of* preserves numerals.

**lemma** *star-of-zero*:  $\text{star-of } 0 = 0$   
 ⟨proof⟩

**lemma** *star-of-one*:  $\text{star-of } 1 = 1$   
 ⟨proof⟩

*star-of* preserves orderings.

**lemma** *star-of-less*:  $(\text{star-of } x < \text{star-of } y) = (x < y)$   
 ⟨proof⟩

**lemma** *star-of-le*:  $(\text{star-of } x \leq \text{star-of } y) = (x \leq y)$   
 ⟨proof⟩

**lemma** *star-of-eq*:  $(\text{star-of } x = \text{star-of } y) = (x = y)$   
 ⟨proof⟩

As above, for 0.

**lemmas** *star-of-0-less* = *star-of-less* [of 0, simplified *star-of-zero*]

**lemmas** *star-of-0-le* = *star-of-le* [of 0, simplified *star-of-zero*]

**lemmas** *star-of-0-eq* = *star-of-eq* [of 0, simplified *star-of-zero*]

**lemmas** *star-of-less-0* = *star-of-less* [of - 0, simplified *star-of-zero*]

**lemmas** *star-of-le-0* = *star-of-le* [of - 0, simplified *star-of-zero*]

**lemmas** *star-of-eq-0* = *star-of-eq* [of - 0, simplified *star-of-zero*]

As above, for 1.

**lemmas** *star-of-1-less* = *star-of-less* [of 1, simplified *star-of-one*]

**lemmas** *star-of-1-le* = *star-of-le* [of 1, simplified *star-of-one*]

**lemmas** *star-of-1-eq* = *star-of-eq* [of 1, simplified *star-of-one*]

**lemmas** *star-of-less-1* = *star-of-less* [of - 1, simplified *star-of-one*]

**lemmas** *star-of-le-1* = *star-of-le* [of - 1, simplified *star-of-one*]

**lemmas** *star-of-eq-1* = *star-of-eq* [of - 1, simplified *star-of-one*]

**lemmas** *star-of-simps* [*simp*] =

```

star-of-add    star-of-diff    star-of-minus
star-of-mult   star-of-divide  star-of-inverse
star-of-mod    star-of-abs
star-of-zero   star-of-one
star-of-less   star-of-le     star-of-eq
star-of-0-less star-of-0-le    star-of-0-eq
star-of-less-0 star-of-le-0    star-of-eq-0
star-of-1-less star-of-1-le    star-of-1-eq
star-of-less-1 star-of-le-1    star-of-eq-1

```

## 2.9 Ordering and lattice classes

**instance** *star* :: (order) order  
 ⟨proof⟩

**instantiation** *star* :: (semilattice-inf) semilattice-inf

**begin**

**definition** *star-inf-def* [transfer-unfold]:  $\text{inf} \equiv *f2* \text{inf}$

**instance** ⟨proof⟩

**end**

**instantiation** *star* :: (semilattice-sup) semilattice-sup

**begin**

**definition** *star-sup-def* [transfer-unfold]:  $\text{sup} \equiv *f2* \text{sup}$

**instance** ⟨proof⟩

**end**

**instance** *star* :: (lattice) lattice ⟨proof⟩

**instance** *star* :: (distrib-lattice) distrib-lattice  
 ⟨proof⟩

**lemma** *Standard-inf* [simp]:  $x \in \text{Standard} \implies y \in \text{Standard} \implies \text{inf } x \ y \in \text{Standard}$   
 ⟨proof⟩

**lemma** *Standard-sup* [simp]:  $x \in \text{Standard} \implies y \in \text{Standard} \implies \text{sup } x \ y \in \text{Standard}$   
 ⟨proof⟩

**lemma** *star-of-inf* [simp]:  $\text{star-of } (\text{inf } x \ y) = \text{inf } (\text{star-of } x) (\text{star-of } y)$   
 ⟨proof⟩

**lemma** *star-of-sup* [simp]:  $\text{star-of } (\text{sup } x \ y) = \text{sup } (\text{star-of } x) (\text{star-of } y)$   
 ⟨proof⟩

**instance** *star* :: (linorder) linorder  
 ⟨proof⟩

**lemma** *star-max-def* [transfer-unfold]:  $\text{max} = *f2* \text{max}$

*<proof>*

**lemma** *star-min-def* [*transfer-unfold*]:  $\text{min} = *f2* \text{min}$   
*<proof>*

**lemma** *Standard-max* [*simp*]:  $x \in \text{Standard} \implies y \in \text{Standard} \implies \text{max } x \ y \in \text{Standard}$   
*<proof>*

**lemma** *Standard-min* [*simp*]:  $x \in \text{Standard} \implies y \in \text{Standard} \implies \text{min } x \ y \in \text{Standard}$   
*<proof>*

**lemma** *star-of-max* [*simp*]:  $\text{star-of } (\text{max } x \ y) = \text{max } (\text{star-of } x) \ (\text{star-of } y)$   
*<proof>*

**lemma** *star-of-min* [*simp*]:  $\text{star-of } (\text{min } x \ y) = \text{min } (\text{star-of } x) \ (\text{star-of } y)$   
*<proof>*

## 2.10 Ordered group classes

**instance** *star* :: (*semigroup-add*) *semigroup-add*  
*<proof>*

**instance** *star* :: (*ab-semigroup-add*) *ab-semigroup-add*  
*<proof>*

**instance** *star* :: (*semigroup-mult*) *semigroup-mult*  
*<proof>*

**instance** *star* :: (*ab-semigroup-mult*) *ab-semigroup-mult*  
*<proof>*

**instance** *star* :: (*comm-monoid-add*) *comm-monoid-add*  
*<proof>*

**instance** *star* :: (*monoid-mult*) *monoid-mult*  
*<proof>*

**instance** *star* :: (*power*) *power* *<proof>*

**instance** *star* :: (*comm-monoid-mult*) *comm-monoid-mult*  
*<proof>*

**instance** *star* :: (*cancel-semigroup-add*) *cancel-semigroup-add*  
*<proof>*

**instance** *star* :: (*cancel-ab-semigroup-add*) *cancel-ab-semigroup-add*  
*<proof>*

**instance** *star* :: (*cancel-comm-monoid-add*) *cancel-comm-monoid-add* ⟨*proof*⟩

**instance** *star* :: (*ab-group-add*) *ab-group-add*  
 ⟨*proof*⟩

**instance** *star* :: (*ordered-ab-semigroup-add*) *ordered-ab-semigroup-add*  
 ⟨*proof*⟩

**instance** *star* :: (*ordered-cancel-ab-semigroup-add*) *ordered-cancel-ab-semigroup-add*  
 ⟨*proof*⟩

**instance** *star* :: (*ordered-ab-semigroup-add-imp-le*) *ordered-ab-semigroup-add-imp-le*  
 ⟨*proof*⟩

**instance** *star* :: (*ordered-comm-monoid-add*) *ordered-comm-monoid-add* ⟨*proof*⟩

**instance** *star* :: (*ordered-ab-semigroup-monoid-add-imp-le*) *ordered-ab-semigroup-monoid-add-imp-le*  
 ⟨*proof*⟩

**instance** *star* :: (*ordered-cancel-comm-monoid-add*) *ordered-cancel-comm-monoid-add*  
 ⟨*proof*⟩

**instance** *star* :: (*ordered-ab-group-add*) *ordered-ab-group-add* ⟨*proof*⟩

**instance** *star* :: (*ordered-ab-group-add-abs*) *ordered-ab-group-add-abs*  
 ⟨*proof*⟩

**instance** *star* :: (*linordered-cancel-ab-semigroup-add*) *linordered-cancel-ab-semigroup-add*  
 ⟨*proof*⟩

## 2.11 Ring and field classes

**instance** *star* :: (*semiring*) *semiring*  
 ⟨*proof*⟩

**instance** *star* :: (*semiring-0*) *semiring-0*  
 ⟨*proof*⟩

**instance** *star* :: (*semiring-0-cancel*) *semiring-0-cancel* ⟨*proof*⟩

**instance** *star* :: (*comm-semiring*) *comm-semiring*  
 ⟨*proof*⟩

**instance** *star* :: (*comm-semiring-0*) *comm-semiring-0* ⟨*proof*⟩

**instance** *star* :: (*comm-semiring-0-cancel*) *comm-semiring-0-cancel* ⟨*proof*⟩

**instance** *star* :: (*zero-neq-one*) *zero-neq-one*  
 ⟨*proof*⟩

**instance** *star* :: (*semiring-1*) *semiring-1* ⟨*proof*⟩

**instance** *star* :: (*comm-semiring-1*) *comm-semiring-1* ⟨*proof*⟩

```

declare dvd-def [transfer-refold]

instance star :: (comm-semiring-1-cancel) comm-semiring-1-cancel
  ⟨proof⟩

instance star :: (semiring-no-zero-divisors) semiring-no-zero-divisors
  ⟨proof⟩

instance star :: (semiring-1-no-zero-divisors) semiring-1-no-zero-divisors ⟨proof⟩

instance star :: (semiring-no-zero-divisors-cancel) semiring-no-zero-divisors-cancel
  ⟨proof⟩

instance star :: (semiring-1-cancel) semiring-1-cancel ⟨proof⟩
instance star :: (ring) ring ⟨proof⟩
instance star :: (comm-ring) comm-ring ⟨proof⟩
instance star :: (ring-1) ring-1 ⟨proof⟩
instance star :: (comm-ring-1) comm-ring-1 ⟨proof⟩
instance star :: (semidom) semidom ⟨proof⟩

instance star :: (semidom-divide) semidom-divide
  ⟨proof⟩

instance star :: (ring-no-zero-divisors) ring-no-zero-divisors ⟨proof⟩
instance star :: (ring-1-no-zero-divisors) ring-1-no-zero-divisors ⟨proof⟩
instance star :: (idom) idom ⟨proof⟩
instance star :: (idom-divide) idom-divide ⟨proof⟩

instance star :: (divide-trivial) divide-trivial
  ⟨proof⟩

instance star :: (division-ring) division-ring
  ⟨proof⟩

instance star :: (field) field
  ⟨proof⟩

instance star :: (ordered-semiring) ordered-semiring
  ⟨proof⟩

instance star :: (ordered-cancel-semiring) ordered-cancel-semiring ⟨proof⟩

instance star :: (linordered-semiring-strict) linordered-semiring-strict
  ⟨proof⟩

instance star :: (ordered-comm-semiring) ordered-comm-semiring
  ⟨proof⟩

```

**instance** *star* :: (*ordered-cancel-comm-semiring*) *ordered-cancel-comm-semiring* ⟨*proof*⟩

**instance** *star* :: (*linordered-comm-semiring-strict*) *linordered-comm-semiring-strict*  
 ⟨*proof*⟩

**instance** *star* :: (*ordered-ring*) *ordered-ring* ⟨*proof*⟩

**instance** *star* :: (*ordered-ring-abs*) *ordered-ring-abs*  
 ⟨*proof*⟩

**instance** *star* :: (*abs-if*) *abs-if*  
 ⟨*proof*⟩

**instance** *star* :: (*linordered-ring-strict*) *linordered-ring-strict* ⟨*proof*⟩

**instance** *star* :: (*ordered-comm-ring*) *ordered-comm-ring* ⟨*proof*⟩

**instance** *star* :: (*linordered-semidom*) *linordered-semidom*  
 ⟨*proof*⟩

**instance** *star* :: (*linordered-idom*) *linordered-idom*  
 ⟨*proof*⟩

**instance** *star* :: (*linordered-field*) *linordered-field* ⟨*proof*⟩

**instance** *star* :: (*algebraic-semidom*) *algebraic-semidom* ⟨*proof*⟩

**instantiation** *star* :: (*normalization-semidom*) *normalization-semidom*  
**begin**

**definition** *unit-factor-star* :: 'a *star* ⇒ 'a *star*  
**where** [*transfer-unfold*]: *unit-factor-star* = \*f\* *unit-factor*

**definition** *normalize-star* :: 'a *star* ⇒ 'a *star*  
**where** [*transfer-unfold*]: *normalize-star* = \*f\* *normalize*

**instance**  
 ⟨*proof*⟩

**end**

**instance** *star* :: (*semidom-modulo*) *semidom-modulo*  
 ⟨*proof*⟩

## 2.12 Power

**lemma** *star-power-def* [*transfer-unfold*]:  $(\wedge) \equiv \lambda x n. (*f* (\lambda x. x \wedge n)) x$   
 ⟨*proof*⟩

**lemma** *Standard-power* [*simp*]:  $x \in \text{Standard} \implies x \wedge n \in \text{Standard}$



*<proof>*

**lemma** *star-of-power* [*simp*]: *star-of* ( $x \wedge n$ ) = *star-of*  $x \wedge n$   
*<proof>*

### 2.13 Number classes

**instance** *star* :: (*numeral*) *numeral* *<proof>*

**lemma** *star-numeral-def* [*transfer-unfold*]: *numeral*  $k$  = *star-of* (*numeral*  $k$ )  
*<proof>*

**lemma** *Standard-numeral* [*simp*]: *numeral*  $k \in$  *Standard*  
*<proof>*

**lemma** *star-of-numeral* [*simp*]: *star-of* (*numeral*  $k$ ) = *numeral*  $k$   
*<proof>*

**lemma** *star-of-nat-def* [*transfer-unfold*]: *of-nat*  $n$  = *star-of* (*of-nat*  $n$ )  
*<proof>*

**lemmas** *star-of-compare-numeral* [*simp*] =  
*star-of-less* [*of numeral k, simplified star-of-numeral*]  
*star-of-le* [*of numeral k, simplified star-of-numeral*]  
*star-of-eq* [*of numeral k, simplified star-of-numeral*]  
*star-of-less* [*of - numeral k, simplified star-of-numeral*]  
*star-of-le* [*of - numeral k, simplified star-of-numeral*]  
*star-of-eq* [*of - numeral k, simplified star-of-numeral*]  
*star-of-less* [*of - numeral k, simplified star-of-numeral*]  
*star-of-le* [*of - numeral k, simplified star-of-numeral*]  
*star-of-eq* [*of - numeral k, simplified star-of-numeral*]  
*star-of-less* [*of - - numeral k, simplified star-of-numeral*]  
*star-of-le* [*of - - numeral k, simplified star-of-numeral*]  
*star-of-eq* [*of - - numeral k, simplified star-of-numeral*] **for**  $k$

**lemma** *Standard-of-nat* [*simp*]: *of-nat*  $n \in$  *Standard*  
*<proof>*

**lemma** *star-of-of-nat* [*simp*]: *star-of* (*of-nat*  $n$ ) = *of-nat*  $n$   
*<proof>*

**lemma** *star-of-int-def* [*transfer-unfold*]: *of-int*  $z$  = *star-of* (*of-int*  $z$ )  
*<proof>*

**lemma** *Standard-of-int* [*simp*]: *of-int*  $z \in$  *Standard*  
*<proof>*

**lemma** *star-of-of-int* [*simp*]: *star-of* (*of-int*  $z$ ) = *of-int*  $z$   
*<proof>*

**instance** *star* :: (*semiring-char-0*) *semiring-char-0*  
 ⟨*proof*⟩

**instance** *star* :: (*ring-char-0*) *ring-char-0* ⟨*proof*⟩

## 2.14 Finite class

**lemma** *starset-finite*: *finite A*  $\implies$  *\*s\** *A* = *star-of* ‘ *A*  
 ⟨*proof*⟩

**instance** *star* :: (*finite*) *finite*  
 ⟨*proof*⟩

**end**

## 3 Hypernatural numbers

**theory** *HyperNat*  
**imports** *StarDef*  
**begin**

**type-synonym** *hypnat* = *nat star*

**abbreviation** *hypnat-of-nat* :: *nat*  $\Rightarrow$  *nat star*  
**where** *hypnat-of-nat*  $\equiv$  *star-of*

**definition** *hSuc* :: *hypnat*  $\Rightarrow$  *hypnat*  
**where** *hSuc-def* [*transfer-unfold*]: *hSuc* = *\*f\** *Suc*

### 3.1 Properties Transferred from Naturals

**lemma** *hSuc-not-zero* [*iff*]:  $\bigwedge m. hSuc\ m \neq 0$   
 ⟨*proof*⟩

**lemma** *zero-not-hSuc* [*iff*]:  $\bigwedge m. 0 \neq hSuc\ m$   
 ⟨*proof*⟩

**lemma** *hSuc-hSuc-eq* [*iff*]:  $\bigwedge m\ n. hSuc\ m = hSuc\ n \longleftrightarrow m = n$   
 ⟨*proof*⟩

**lemma** *zero-less-hSuc* [*iff*]:  $\bigwedge n. 0 < hSuc\ n$   
 ⟨*proof*⟩

**lemma** *hypnat-minus-zero* [*simp*]:  $\bigwedge z::hypnat. z - z = 0$   
 ⟨*proof*⟩

**lemma** *hypnat-diff-0-eq-0* [*simp*]:  $\bigwedge n::hypnat. 0 - n = 0$   
 ⟨*proof*⟩

**lemma** *hypnat-add-is-0* [iff]:  $\bigwedge m n::hypnat. m + n = 0 \longleftrightarrow m = 0 \wedge n = 0$   
 ⟨proof⟩

**lemma** *hypnat-diff-diff-left*:  $\bigwedge i j k::hypnat. i - j - k = i - (j + k)$   
 ⟨proof⟩

**lemma** *hypnat-diff-commute*:  $\bigwedge i j k::hypnat. i - j - k = i - k - j$   
 ⟨proof⟩

**lemma** *hypnat-diff-add-inverse* [simp]:  $\bigwedge m n::hypnat. n + m - n = m$   
 ⟨proof⟩

**lemma** *hypnat-diff-add-inverse2* [simp]:  $\bigwedge m n::hypnat. m + n - n = m$   
 ⟨proof⟩

**lemma** *hypnat-diff-cancel* [simp]:  $\bigwedge k m n::hypnat. (k + m) - (k + n) = m - n$   
 ⟨proof⟩

**lemma** *hypnat-diff-cancel2* [simp]:  $\bigwedge k m n::hypnat. (m + k) - (n + k) = m - n$   
 ⟨proof⟩

**lemma** *hypnat-diff-add-0* [simp]:  $\bigwedge m n::hypnat. n - (n + m) = 0$   
 ⟨proof⟩

**lemma** *hypnat-diff-mult-distrib*:  $\bigwedge k m n::hypnat. (m - n) * k = (m * k) - (n * k)$   
 ⟨proof⟩

**lemma** *hypnat-diff-mult-distrib2*:  $\bigwedge k m n::hypnat. k * (m - n) = (k * m) - (k * n)$   
 ⟨proof⟩

**lemma** *hypnat-le-zero-cancel* [iff]:  $\bigwedge n::hypnat. n \leq 0 \longleftrightarrow n = 0$   
 ⟨proof⟩

**lemma** *hypnat-mult-is-0* [simp]:  $\bigwedge m n::hypnat. m * n = 0 \longleftrightarrow m = 0 \vee n = 0$   
 ⟨proof⟩

**lemma** *hypnat-diff-is-0-eq* [simp]:  $\bigwedge m n::hypnat. m - n = 0 \longleftrightarrow m \leq n$   
 ⟨proof⟩

**lemma** *hypnat-not-less0* [iff]:  $\bigwedge n::hypnat. \neg n < 0$   
 ⟨proof⟩

**lemma** *hypnat-less-one* [iff]:  $\bigwedge n::hypnat. n < 1 \longleftrightarrow n = 0$   
 ⟨proof⟩

**lemma** *hypnat-add-diff-inverse*:  $\bigwedge m n::hypnat. \neg m < n \implies n + (m - n) = m$

*<proof>*

**lemma** *hypnat-le-add-diff-inverse* [*simp*]:  $\bigwedge m n :: \text{hypnat}. n \leq m \implies n + (m - n) = m$   
*<proof>*

**lemma** *hypnat-le-add-diff-inverse2* [*simp*]:  $\bigwedge m n :: \text{hypnat}. n \leq m \implies (m - n) + n = m$   
*<proof>*

**declare** *hypnat-le-add-diff-inverse2* [*OF order-less-imp-le*]

**lemma** *hypnat-le0* [*iff*]:  $\bigwedge n :: \text{hypnat}. 0 \leq n$   
*<proof>*

**lemma** *hypnat-le-add1* [*simp*]:  $\bigwedge x n :: \text{hypnat}. x \leq x + n$   
*<proof>*

**lemma** *hypnat-add-self-le* [*simp*]:  $\bigwedge x n :: \text{hypnat}. x \leq n + x$   
*<proof>*

**lemma** *hypnat-add-one-self-less* [*simp*]:  $x < x + 1$  **for**  $x :: \text{hypnat}$   
*<proof>*

**lemma** *hypnat-neq0-conv* [*iff*]:  $\bigwedge n :: \text{hypnat}. n \neq 0 \iff 0 < n$   
*<proof>*

**lemma** *hypnat-gt-zero-iff*:  $0 < n \iff 1 \leq n$  **for**  $n :: \text{hypnat}$   
*<proof>*

**lemma** *hypnat-gt-zero-iff2*:  $0 < n \iff (\exists m. n = m + 1)$  **for**  $n :: \text{hypnat}$   
*<proof>*

**lemma** *hypnat-add-self-not-less*:  $\neg x + y < x$  **for**  $x y :: \text{hypnat}$   
*<proof>*

**lemma** *hypnat-diff-split*:  $P (a - b) \iff (a < b \implies P 0) \wedge (\forall d. a = b + d \implies P d)$   
**for**  $a b :: \text{hypnat}$   
 — elimination of  $-$  on *hypnat*  
*<proof>*

### 3.2 Properties of the set of embedded natural numbers

**lemma** *of-nat-eq-star-of* [*simp*]: *of-nat* = *star-of*  
*<proof>*

**lemma** *Nats-eq-Standard*:  $(\text{Nats} :: \text{nat star set}) = \text{Standard}$   
*<proof>*

**lemma** *hypnat-of-nat-mem-Nats* [simp]: *hypnat-of-nat*  $n \in \text{Nats}$   
 ⟨proof⟩

**lemma** *hypnat-of-nat-one* [simp]: *hypnat-of-nat* (*Suc* 0) = 1  
 ⟨proof⟩

**lemma** *hypnat-of-nat-Suc* [simp]: *hypnat-of-nat* (*Suc*  $n$ ) = *hypnat-of-nat*  $n + 1$   
 ⟨proof⟩

**lemma** *of-nat-eq-add*:  
 fixes  $d :: \text{hypnat}$   
 shows *of-nat*  $m = \text{of-nat } n + d \implies d \in \text{range of-nat}$   
 ⟨proof⟩

**lemma** *Nats-diff* [simp]:  $a \in \text{Nats} \implies b \in \text{Nats} \implies a - b \in \text{Nats}$  for  $a \ b :: \text{hypnat}$   
 ⟨proof⟩

### 3.3 Infinite Hypernatural Numbers – *HNatInfinite*

The set of infinite hypernatural numbers.

**definition** *HNatInfinite* :: *hypnat set*  
 where *HNatInfinite* = { $n. n \notin \text{Nats}$ }

**lemma** *Nats-not-HNatInfinite-iff*:  $x \in \text{Nats} \longleftrightarrow x \notin \text{HNatInfinite}$   
 ⟨proof⟩

**lemma** *HNatInfinite-not-Nats-iff*:  $x \in \text{HNatInfinite} \longleftrightarrow x \notin \text{Nats}$   
 ⟨proof⟩

**lemma** *star-of-neq-HNatInfinite*:  $N \in \text{HNatInfinite} \implies \text{star-of } n \neq N$   
 ⟨proof⟩

**lemma** *star-of-Suc-lessI*:  $\bigwedge N. \text{star-of } n < N \implies \text{star-of } (\text{Suc } n) \neq N \implies \text{star-of } (\text{Suc } n) < N$   
 ⟨proof⟩

**lemma** *star-of-less-HNatInfinite*:  
 assumes  $N: N \in \text{HNatInfinite}$   
 shows *star-of*  $n < N$   
 ⟨proof⟩

**lemma** *star-of-le-HNatInfinite*:  $N \in \text{HNatInfinite} \implies \text{star-of } n \leq N$   
 ⟨proof⟩

#### 3.3.1 Closure Rules

**lemma** *Nats-less-HNatInfinite*:  $x \in \text{Nats} \implies y \in \text{HNatInfinite} \implies x < y$

*<proof>*

**lemma** *Nats-le-HNatInfinite*:  $x \in \text{Nats} \implies y \in \text{HNatInfinite} \implies x \leq y$   
*<proof>*

**lemma** *zero-less-HNatInfinite*:  $x \in \text{HNatInfinite} \implies 0 < x$   
*<proof>*

**lemma** *one-less-HNatInfinite*:  $x \in \text{HNatInfinite} \implies 1 < x$   
*<proof>*

**lemma** *one-le-HNatInfinite*:  $x \in \text{HNatInfinite} \implies 1 \leq x$   
*<proof>*

**lemma** *zero-not-mem-HNatInfinite* [*simp*]:  $0 \notin \text{HNatInfinite}$   
*<proof>*

**lemma** *Nats-downward-closed*:  $x \in \text{Nats} \implies y \leq x \implies y \in \text{Nats}$  **for**  $x\ y :: \text{hypnat}$   
*<proof>*

**lemma** *HNatInfinite-upward-closed*:  $x \in \text{HNatInfinite} \implies x \leq y \implies y \in \text{HNatInfinite}$   
*<proof>*

**lemma** *HNatInfinite-add*:  $x \in \text{HNatInfinite} \implies x + y \in \text{HNatInfinite}$   
*<proof>*

**lemma** *HNatInfinite-diff*:  $\llbracket x \in \text{HNatInfinite}; y \in \text{Nats} \rrbracket \implies x - y \in \text{HNatInfinite}$   
*<proof>*

**lemma** *HNatInfinite-is-Suc*:  $x \in \text{HNatInfinite} \implies \exists y. x = y + 1$  **for**  $x :: \text{hypnat}$   
*<proof>*

### 3.4 Existence of an infinite hypernatural number

$\omega$  is in fact an infinite hypernatural number = [ $\langle 1, 2, 3, \dots \rangle$ ]

**definition** *whn* :: *hypnat*

**where** *hypnat-omega-def*:  $\text{whn} = \text{star-n } (\lambda n :: \text{nat}. n)$

**lemma** *hypnat-of-nat-neq-whn*:  $\text{hypnat-of-nat } n \neq \text{whn}$   
*<proof>*

**lemma** *whn-neq-hypnat-of-nat*:  $\text{whn} \neq \text{hypnat-of-nat } n$   
*<proof>*

**lemma** *whn-not-Nats* [*simp*]:  $\text{whn} \notin \text{Nats}$   
*<proof>*

**lemma** *HNatInfinite-whn* [*simp*]:  $\text{whn} \in \text{HNatInfinite}$

*<proof>*

**lemma** *lemma-unbounded-set* [*simp*]: *eventually*  $(\lambda n::nat. m < n) \mathcal{U}$   
*<proof>*

**lemma** *hypnat-of-nat-eq*: *hypnat-of-nat*  $m = star-n (\lambda n::nat. m)$   
*<proof>*

**lemma** *SHNat-eq*:  $Nats = \{n. \exists N. n = hypnat-of-nat N\}$   
*<proof>*

**lemma** *Nats-less-whn*:  $n \in Nats \implies n < whn$   
*<proof>*

**lemma** *Nats-le-whn*:  $n \in Nats \implies n \leq whn$   
*<proof>*

**lemma** *hypnat-of-nat-less-whn* [*simp*]: *hypnat-of-nat*  $n < whn$   
*<proof>*

**lemma** *hypnat-of-nat-le-whn* [*simp*]: *hypnat-of-nat*  $n \leq whn$   
*<proof>*

**lemma** *hypnat-zero-less-hypnat-omega* [*simp*]:  $0 < whn$   
*<proof>*

**lemma** *hypnat-one-less-hypnat-omega* [*simp*]:  $1 < whn$   
*<proof>*

### 3.4.1 Alternative characterization of the set of infinite hypernaturals

$HNatInfinite = \{N. \forall n \in \mathbb{N}. n < N\}$

unused, but possibly interesting

**lemma** *HNatInfinite-FreeUltrafilterNat-eventually*:  
**assumes**  $\bigwedge k::nat. eventually (\lambda n. f n \neq k) \mathcal{U}$   
**shows** *eventually*  $(\lambda n. m < f n) \mathcal{U}$   
*<proof>*

**lemma** *HNatInfinite-iff*:  $HNatInfinite = \{N. \forall n \in Nats. n < N\}$   
*<proof>*

### 3.4.2 Alternative Characterization of *HNatInfinite* using Free Ultrafilter

**lemma** *HNatInfinite-FreeUltrafilterNat*:  
 $star-n X \in HNatInfinite \implies \forall u. eventually (\lambda n. u < X n) \mathcal{U}$   
*<proof>*

**lemma** *FreeUltrafilterNat-HNatInfinite*:

$\forall u. \text{eventually } (\lambda n. u < X n) \mathcal{U} \implies \text{star-}n X \in \text{HNatInfinite}$   
 ⟨proof⟩

**lemma** *HNatInfinite-FreeUltrafilterNat-iff*:

$(\text{star-}n X \in \text{HNatInfinite}) = (\forall u. \text{eventually } (\lambda n. u < X n) \mathcal{U})$   
 ⟨proof⟩

### 3.5 Embedding of the Hypernaturals into other types

**definition** *of-hypnat* :: *hypnat*  $\Rightarrow$  'a::semiring-1-cancel star

where *of-hypnat-def* [*transfer-unfold*]: *of-hypnat* = \*f\* *of-nat*

**lemma** *of-hypnat-0* [*simp*]: *of-hypnat* 0 = 0

⟨proof⟩

**lemma** *of-hypnat-1* [*simp*]: *of-hypnat* 1 = 1

⟨proof⟩

**lemma** *of-hypnat-hSuc*:  $\bigwedge m. \text{of-hypnat } (hSuc m) = 1 + \text{of-hypnat } m$

⟨proof⟩

**lemma** *of-hypnat-add* [*simp*]:  $\bigwedge m n. \text{of-hypnat } (m + n) = \text{of-hypnat } m + \text{of-hypnat } n$

⟨proof⟩

**lemma** *of-hypnat-mult* [*simp*]:  $\bigwedge m n. \text{of-hypnat } (m * n) = \text{of-hypnat } m * \text{of-hypnat } n$

⟨proof⟩

**lemma** *of-hypnat-less-iff* [*simp*]:

$\bigwedge m n. \text{of-hypnat } m < (\text{of-hypnat } n :: 'a :: \text{linordered-semidom star}) \longleftrightarrow m < n$   
 ⟨proof⟩

**lemma** *of-hypnat-0-less-iff* [*simp*]:

$\bigwedge n. 0 < (\text{of-hypnat } n :: 'a :: \text{linordered-semidom star}) \longleftrightarrow 0 < n$   
 ⟨proof⟩

**lemma** *of-hypnat-less-0-iff* [*simp*]:  $\bigwedge m. \neg (\text{of-hypnat } m :: 'a :: \text{linordered-semidom star}) < 0$

⟨proof⟩

**lemma** *of-hypnat-le-iff* [*simp*]:

$\bigwedge m n. \text{of-hypnat } m \leq (\text{of-hypnat } n :: 'a :: \text{linordered-semidom star}) \longleftrightarrow m \leq n$   
 ⟨proof⟩

**lemma** *of-hypnat-0-le-iff* [*simp*]:  $\bigwedge n. 0 \leq (\text{of-hypnat } n :: 'a :: \text{linordered-semidom star})$



*<proof>*

**lemma** *of-hypnat-le-0-iff* [simp]:  $\bigwedge m. (\text{of-hypnat } m :: 'a::\text{linordered-semidom star}) \leq 0 \longleftrightarrow m = 0$   
*<proof>*

**lemma** *of-hypnat-eq-iff* [simp]:  
 $\bigwedge m n. \text{of-hypnat } m = (\text{of-hypnat } n :: 'a::\text{linordered-semidom star}) \longleftrightarrow m = n$   
*<proof>*

**lemma** *of-hypnat-eq-0-iff* [simp]:  $\bigwedge m. (\text{of-hypnat } m :: 'a::\text{linordered-semidom star}) = 0 \longleftrightarrow m = 0$   
*<proof>*

**lemma** *HNatInfinite-of-hypnat-gt-zero*:  
 $N \in \text{HNatInfinite} \implies (0 :: 'a::\text{linordered-semidom star}) < \text{of-hypnat } N$   
*<proof>*

end

## 4 Construction of Hyperreals Using Ultrafilters

**theory** *HyperDef*

**imports** *Complex-Main HyperNat*

**begin**

**type-synonym** *hypreal* = *real star*

**abbreviation** *hypreal-of-real* :: *real*  $\Rightarrow$  *real star*  
**where** *hypreal-of-real*  $\equiv$  *star-of*

**abbreviation** *hypreal-of-hypnat* :: *hypnat*  $\Rightarrow$  *hypreal*  
**where** *hypreal-of-hypnat*  $\equiv$  *of-hypnat*

**definition** *omega* :: *hypreal* ( $\omega$ )  
**where**  $\omega = \text{star-n } (\lambda n. \text{real } (\text{Suc } n))$   
— an infinite number = [*1, 2, 3, ...*]

**definition** *epsilon* :: *hypreal* ( $\varepsilon$ )  
**where**  $\varepsilon = \text{star-n } (\lambda n. \text{inverse } (\text{real } (\text{Suc } n)))$   
— an infinitesimal number = [*1, 1/2, 1/3, ...*]

### 4.1 Real vector class instances

**instantiation** *star* :: (*scaleR*) *scaleR*

**begin**

**definition** *star-scaleR-def* [transfer-unfold]: *scaleR* *r*  $\equiv$  *\*f\** (*scaleR* *r*)

**instance** *<proof>*

**end**

**lemma** *Standard-scaleR* [simp]:  $x \in \text{Standard} \implies \text{scaleR } r \ x \in \text{Standard}$   
 ⟨proof⟩

**lemma** *star-of-scaleR* [simp]:  $\text{star-of } (\text{scaleR } r \ x) = \text{scaleR } r \ (\text{star-of } x)$   
 ⟨proof⟩

**instance** *star* :: (real-vector) real-vector  
 ⟨proof⟩

**instance** *star* :: (real-algebra) real-algebra  
 ⟨proof⟩

**instance** *star* :: (real-algebra-1) real-algebra-1 ⟨proof⟩

**instance** *star* :: (real-div-algebra) real-div-algebra ⟨proof⟩

**instance** *star* :: (field-char-0) field-char-0 ⟨proof⟩

**instance** *star* :: (real-field) real-field ⟨proof⟩

**lemma** *star-of-real-def* [transfer-unfold]:  $\text{of-real } r = \text{star-of } (\text{of-real } r)$   
 ⟨proof⟩

**lemma** *Standard-of-real* [simp]:  $\text{of-real } r \in \text{Standard}$   
 ⟨proof⟩

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

**lemma** *of-real-eq-star-of* [simp]:  $\text{of-real} = \text{star-of}$   
 ⟨proof⟩

**lemma** *Reals-eq-Standard*:  $(\mathbb{R} :: \text{hypreal set}) = \text{Standard}$   
 ⟨proof⟩

## 4.2 Injection from hypreal

**definition** *of-hypreal* :: hypreal  $\Rightarrow$  'a::real-algebra-1 *star*  
 where [transfer-unfold]:  $\text{of-hypreal} = *f* \ \text{of-real}$

**lemma** *Standard-of-hypreal* [simp]:  $r \in \text{Standard} \implies \text{of-hypreal } r \in \text{Standard}$   
 ⟨proof⟩

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

**lemma** *of-hypreal-1* [simp]:  $\text{of-hypreal } 1 = 1$   
 ⟨proof⟩

**lemma** *of-hypreal-add* [simp]:  $\bigwedge x y. \text{of-hypreal } (x + y) = \text{of-hypreal } x + \text{of-hypreal } y$   
 ⟨proof⟩

**lemma** *of-hypreal-minus* [simp]:  $\bigwedge x. \text{of-hypreal } (- x) = - \text{of-hypreal } x$   
 ⟨proof⟩

**lemma** *of-hypreal-diff* [simp]:  $\bigwedge x y. \text{of-hypreal } (x - y) = \text{of-hypreal } x - \text{of-hypreal } y$   
 ⟨proof⟩

**lemma** *of-hypreal-mult* [simp]:  $\bigwedge x y. \text{of-hypreal } (x * y) = \text{of-hypreal } x * \text{of-hypreal } y$   
 ⟨proof⟩

**lemma** *of-hypreal-inverse* [simp]:  
 $\bigwedge x. \text{of-hypreal } (\text{inverse } x) =$   
 $\text{inverse } (\text{of-hypreal } x :: 'a::\{\text{real-div-algebra, division-ring}\} \text{star})$   
 ⟨proof⟩

**lemma** *of-hypreal-divide* [simp]:  
 $\bigwedge x y. \text{of-hypreal } (x / y) =$   
 $(\text{of-hypreal } x / \text{of-hypreal } y :: 'a::\{\text{real-field, field}\} \text{star})$   
 ⟨proof⟩

**lemma** *of-hypreal-eq-iff* [simp]:  $\bigwedge x y. (\text{of-hypreal } x = \text{of-hypreal } y) = (x = y)$   
 ⟨proof⟩

**lemma** *of-hypreal-eq-0-iff* [simp]:  $\bigwedge x. (\text{of-hypreal } x = 0) = (x = 0)$   
 ⟨proof⟩

### 4.3 Properties of *starrel*

**lemma** *lemma-starrel-refl* [simp]:  $x \in \text{starrel } \{x\}$   
 ⟨proof⟩

**lemma** *starrel-in-hypreal* [simp]:  $\text{starrel } \{x\} \in \text{star}$   
 ⟨proof⟩

**declare** *Abs-star-inject* [simp] *Abs-star-inverse* [simp]  
**declare** *equiv-starrel* [THEN *eq-equiv-class-iff*, simp]

### 4.4 *hypreal-of-real*: the Injection from *real* to *hypreal*

**lemma** *inj-star-of*: *inj star-of*  
 ⟨proof⟩

**lemma** *mem-Rep-star-iff*:  $X \in \text{Rep-star } x \longleftrightarrow x = \text{star-n } X$   
 ⟨proof⟩

**lemma** *Rep-star-star-n-iff* [simp]:  $X \in \text{Rep-star} (\text{star-n } Y) \longleftrightarrow \text{eventually } (\lambda n. Y\ n = X\ n) \mathcal{U}$   
 ⟨proof⟩

**lemma** *Rep-star-star-n*:  $X \in \text{Rep-star} (\text{star-n } X)$   
 ⟨proof⟩

#### 4.5 Properties of *star-n*

**lemma** *star-n-add*:  $\text{star-n } X + \text{star-n } Y = \text{star-n } (\lambda n. X\ n + Y\ n)$   
 ⟨proof⟩

**lemma** *star-n-minus*:  $-\text{star-n } X = \text{star-n } (\lambda n. -(X\ n))$   
 ⟨proof⟩

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

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

**lemma** *star-n-inverse*:  $\text{inverse} (\text{star-n } X) = \text{star-n } (\lambda n. \text{inverse} (X\ n))$   
 ⟨proof⟩

**lemma** *star-n-le*:  $\text{star-n } X \leq \text{star-n } Y = \text{eventually } (\lambda n. X\ n \leq Y\ n) \mathcal{U}$   
 ⟨proof⟩

**lemma** *star-n-less*:  $\text{star-n } X < \text{star-n } Y = \text{eventually } (\lambda n. X\ n < Y\ n) \mathcal{U}$   
 ⟨proof⟩

**lemma** *star-n-zero-num*:  $0 = \text{star-n } (\lambda n. 0)$   
 ⟨proof⟩

**lemma** *star-n-one-num*:  $1 = \text{star-n } (\lambda n. 1)$   
 ⟨proof⟩

**lemma** *star-n-abs*:  $|\text{star-n } X| = \text{star-n } (\lambda n. |X\ n|)$   
 ⟨proof⟩

**lemma** *hypreal-omega-gt-zero* [simp]:  $0 < \omega$   
 ⟨proof⟩

#### 4.6 Existence of Infinite Hyperreal Number

Existence of infinite number not corresponding to any real number. Use assumption that member  $\mathcal{U}$  is not finite.

**lemma** *hypreal-of-real-not-eq-omega*:  $\text{hypreal-of-real } x \neq \omega$   
 ⟨proof⟩

Existence of infinitesimal number also not corresponding to any real number.

**lemma** *hypreal-of-real-not-eq-epsilon*: *hypreal-of-real*  $x \neq \varepsilon$   
 ⟨*proof*⟩

**lemma** *epsilon-ge-zero* [*simp*]:  $0 \leq \varepsilon$   
 ⟨*proof*⟩

**lemma** *epsilon-not-zero*:  $\varepsilon \neq 0$   
 ⟨*proof*⟩

**lemma** *epsilon-inverse-omega*:  $\varepsilon = \text{inverse } \omega$   
 ⟨*proof*⟩

**lemma** *epsilon-gt-zero*:  $0 < \varepsilon$   
 ⟨*proof*⟩

## 4.7 Embedding the Naturals into the Hyperreals

**abbreviation** *hypreal-of-nat* :: *nat*  $\Rightarrow$  *hypreal*  
 where *hypreal-of-nat*  $\equiv$  *of-nat*

**lemma** *SNat-eq*:  $\text{Nats} = \{n. \exists N. n = \text{hypreal-of-nat } N\}$   
 ⟨*proof*⟩

Naturals embedded in hyperreals: is a hyperreal c.f. NS extension.

**lemma** *hypreal-of-nat*: *hypreal-of-nat*  $m = \text{star-n } (\lambda n. \text{real } m)$   
 ⟨*proof*⟩

⟨*ML*⟩

## 4.8 Exponentials on the Hyperreals

**lemma** *hpowr-0* [*simp*]:  $r \wedge 0 = (1::\text{hypreal})$   
 for  $r :: \text{hypreal}$   
 ⟨*proof*⟩

**lemma** *hpowr-Suc* [*simp*]:  $r \wedge (\text{Suc } n) = r * (r \wedge n)$   
 for  $r :: \text{hypreal}$   
 ⟨*proof*⟩

**lemma** *hrealpow*:  $\text{star-n } X \wedge m = \text{star-n } (\lambda n. (X \text{ n}::\text{real}) \wedge m)$   
 ⟨*proof*⟩

**lemma** *hrealpow-sum-square-expand*:  
 $(x + y) \wedge \text{Suc } (\text{Suc } 0) =$   
 $x \wedge \text{Suc } (\text{Suc } 0) + y \wedge \text{Suc } (\text{Suc } 0) + (\text{hypreal-of-nat } (\text{Suc } (\text{Suc } 0))) * x * y$   
 for  $x \ y :: \text{hypreal}$   
 ⟨*proof*⟩

**lemma** *power-hypreal-of-real-numeral*:

$$(\text{numeral } v :: \text{hypreal}) \wedge n = \text{hypreal-of-real } ((\text{numeral } v) \wedge n)$$

*<proof>*

**declare** *power-hypreal-of-real-numeral* [*of - numeral w, simp*] **for** *w*

**lemma** *power-hypreal-of-real-neg-numeral*:

$$(- \text{numeral } v :: \text{hypreal}) \wedge n = \text{hypreal-of-real } ((- \text{numeral } v) \wedge n)$$

*<proof>*

**declare** *power-hypreal-of-real-neg-numeral* [*of - numeral w, simp*] **for** *w*

## 4.9 Powers with Hypernatural Exponents

Hypernatural powers of hyperreals.

**definition** *pow* :: 'a::power star  $\Rightarrow$  nat star  $\Rightarrow$  'a star (**infixr** *pow* 80)

**where** *hyperpow-def* [*transfer-unfold*]:  $R \text{ pow } N = (*f2* (\wedge)) R N$

**lemma** *Standard-hyperpow* [*simp*]:  $r \in \text{Standard} \Longrightarrow n \in \text{Standard} \Longrightarrow r \text{ pow } n \in \text{Standard}$

*<proof>*

**lemma** *hyperpow*:  $\text{star-n } X \text{ pow star-n } Y = \text{star-n } (\lambda n. X n \wedge Y n)$

*<proof>*

**lemma** *hyperpow-zero* [*simp*]:  $\bigwedge n. (0 :: 'a :: \{\text{power, semiring-0}\} \text{star}) \text{ pow } (n + (1 :: \text{hypnat})) = 0$

*<proof>*

**lemma** *hyperpow-not-zero*:  $\bigwedge r n. r \neq (0 :: 'a :: \{\text{field}\} \text{star}) \Longrightarrow r \text{ pow } n \neq 0$

*<proof>*

**lemma** *hyperpow-inverse*:  $\bigwedge r n. r \neq (0 :: 'a :: \{\text{field}\} \text{star}) \Longrightarrow \text{inverse } (r \text{ pow } n) = (\text{inverse } r) \text{ pow } n$

*<proof>*

**lemma** *hyperpow-hrabs*:  $\bigwedge r n. |r :: 'a :: \{\text{linordered-idom}\} \text{star}| \text{ pow } n = |r \text{ pow } n|$

*<proof>*

**lemma** *hyperpow-add*:  $\bigwedge r n m. (r :: 'a :: \{\text{monoid-mult}\} \text{star}) \text{ pow } (n + m) = (r \text{ pow } n) * (r \text{ pow } m)$

*<proof>*

**lemma** *hyperpow-one* [*simp*]:  $\bigwedge r. (r :: 'a :: \{\text{monoid-mult}\} \text{star}) \text{ pow } (1 :: \text{hypnat}) = r$

*<proof>*

**lemma** *hyperpow-two*:  $\bigwedge r. (r :: 'a :: \{\text{monoid-mult}\} \text{star}) \text{ pow } (2 :: \text{hypnat}) = r * r$

*<proof>*

**lemma** *hyperpow-gt-zero*:  $\bigwedge r n. (0 :: 'a :: \{\text{linordered-semidom}\} \text{star}) < r \Longrightarrow 0 < r \text{ pow } n$

*<proof>*

**lemma** *hyperpow-ge-zero*:  $\bigwedge r n. (0::'a::\{\text{linordered-semidom}\} \text{star}) \leq r \implies 0 \leq r \text{ pow } n$   
*<proof>*

**lemma** *hyperpow-le*:  $\bigwedge x y n. (0::'a::\{\text{linordered-semidom}\} \text{star}) < x \implies x \leq y \implies x \text{ pow } n \leq y \text{ pow } n$   
*<proof>*

**lemma** *hyperpow-eq-one* [simp]:  $\bigwedge n. 1 \text{ pow } n = (1::'a::\{\text{monoid-mult star}\})$   
*<proof>*

**lemma** *hrabs-hyperpow-minus* [simp]:  $\bigwedge (a::'a::\{\text{linordered-idom star}\}) n. |(-a) \text{ pow } n| = |a \text{ pow } n|$   
*<proof>*

**lemma** *hyperpow-mult*:  $\bigwedge r s n. (r * s::'a::\{\text{comm-monoid-mult star}\}) \text{ pow } n = (r \text{ pow } n) * (s \text{ pow } n)$   
*<proof>*

**lemma** *hyperpow-two-le* [simp]:  $\bigwedge r. (0::'a::\{\text{monoid-mult, linordered-ring-strict}\} \text{star}) \leq r \text{ pow } 2$   
*<proof>*

**lemma** *hyperpow-two-hrabs* [simp]:  $|x::'a::\{\text{linordered-idom star}\}| \text{ pow } 2 = x \text{ pow } 2$   
*<proof>*

**lemma** *hyperpow-two-gt-one*:  $\bigwedge r::'a::\{\text{linordered-semidom star}\}. 1 < r \implies 1 < r \text{ pow } 2$   
*<proof>*

**lemma** *hyperpow-two-ge-one*:  $\bigwedge r::'a::\{\text{linordered-semidom star}\}. 1 \leq r \implies 1 \leq r \text{ pow } 2$   
*<proof>*

**lemma** *two-hyperpow-ge-one* [simp]:  $(1::\text{hypreal}) \leq 2 \text{ pow } n$   
*<proof>*

**lemma** *hyperpow-minus-one2* [simp]:  $\bigwedge n. (-1) \text{ pow } (2 * n) = (1::\text{hypreal})$   
*<proof>*

**lemma** *hyperpow-less-le*:  $\bigwedge r n N. (0::\text{hypreal}) \leq r \implies r \leq 1 \implies n < N \implies r \text{ pow } N \leq r \text{ pow } n$   
*<proof>*

**lemma** *hyperpow-SHNat-le*:  
 $0 \leq r \implies r \leq (1::\text{hypreal}) \implies N \in \text{HNatInfinite} \implies \forall n \in \text{Nats}. r \text{ pow } N \leq r \text{ pow } n$

*<proof>*

**lemma** *hyperpow-realpow*: (*hypreal-of-real*  $r$ ) *pow* (*hypnat-of-nat*  $n$ ) = *hypreal-of-real* ( $r \hat{\ } n$ )  
*<proof>*

**lemma** *hyperpow-SReal [simp]*: (*hypreal-of-real*  $r$ ) *pow* (*hypnat-of-nat*  $n$ )  $\in \mathbb{R}$   
*<proof>*

**lemma** *hyperpow-zero-HNatInfinite [simp]*:  $N \in \text{HNatInfinite} \implies (0::\text{hypreal}) \text{ pow } N = 0$   
*<proof>*

**lemma** *hyperpow-le-le*:  $(0::\text{hypreal}) \leq r \implies r \leq 1 \implies n \leq N \implies r \text{ pow } N \leq r \text{ pow } n$   
*<proof>*

**lemma** *hyperpow-Suc-le-self2*:  $(0::\text{hypreal}) \leq r \implies r < 1 \implies r \text{ pow } (n + (1::\text{hypnat})) \leq r \text{ pow } n$   
*<proof>*

**lemma** *hyperpow-hypnat-of-nat*:  $\bigwedge x. x \text{ pow hypnat-of-nat } n = x \hat{\ } n$   
*<proof>*

**lemma** *of-hypreal-hyperpow*:  
 $\bigwedge x n. \text{of-hypreal } (x \text{ pow } n) = (\text{of-hypreal } x::'a::\{\text{real-algebra-1}\} \text{ star}) \text{ pow } n$   
*<proof>*

end

## 5 Infinite Numbers, Infinitesimals, Infinitely Close Relation

**theory** *NSA*  
**imports** *HyperDef HOL-Library.Lub-Glb*  
**begin**

**definition** *hnorm* :: '*a*::*real-normed-vector star*  $\implies$  *real star*  
**where** [*transfer-unfold*]: *hnorm* = *\*f\* norm*

**definition** *Infinitesimal* :: '*a*::*real-normed-vector*) *star set*  
**where** *Infinitesimal* =  $\{x. \forall r \in \text{Reals}. 0 < r \longrightarrow \text{hnorm } x < r\}$

**definition** *HFinite* :: '*a*::*real-normed-vector*) *star set*  
**where** *HFinite* =  $\{x. \exists r \in \text{Reals}. \text{hnorm } x < r\}$

**definition** *HInfinite* :: '*a*::*real-normed-vector*) *star set*  
**where** *HInfinite* =  $\{x. \forall r \in \text{Reals}. r < \text{hnorm } x\}$



**definition** *approx* :: 'a::real-normed-vector star  $\Rightarrow$  'a star  $\Rightarrow$  bool (infixl  $\approx$  50)

**where**  $x \approx y \iff x - y \in \text{Infinitesimal}$   
 — the “infinitely close” relation

**definition** *st* :: hypreal  $\Rightarrow$  hypreal

**where**  $st = (\lambda x. \text{SOME } r. x \in \text{HFinite} \wedge r \in \mathbb{R} \wedge r \approx x)$   
 — the standard part of a hyperreal

**definition** *monad* :: 'a::real-normed-vector star  $\Rightarrow$  'a star set

**where**  $monad\ x = \{y. x \approx y\}$

**definition** *galaxy* :: 'a::real-normed-vector star  $\Rightarrow$  'a star set

**where**  $galaxy\ x = \{y. (x + -y) \in \text{HFinite}\}$

**lemma** *SReal-def*:  $\mathbb{R} \equiv \{x. \exists r. x = \text{hypreal-of-real } r\}$

*<proof>*

## 5.1 Nonstandard Extension of the Norm Function

**definition** *scaleHR* :: real star  $\Rightarrow$  'a star  $\Rightarrow$  'a::real-normed-vector star

**where** [*transfer-unfold*]:  $scaleHR = \text{starfun2 } scaleR$

**lemma** *Standard-hnorm* [*simp*]:  $x \in \text{Standard} \implies \text{hnorm } x \in \text{Standard}$

*<proof>*

**lemma** *star-of-norm* [*simp*]:  $\text{star-of } (\text{norm } x) = \text{hnorm } (\text{star-of } x)$

*<proof>*

**lemma** *hnorm-ge-zero* [*simp*]:  $\bigwedge x::'a::\text{real-normed-vector star}. 0 \leq \text{hnorm } x$

*<proof>*

**lemma** *hnorm-eq-zero* [*simp*]:  $\bigwedge x::'a::\text{real-normed-vector star}. \text{hnorm } x = 0 \iff x = 0$

*<proof>*

**lemma** *hnorm-triangle-ineq*:  $\bigwedge x\ y::'a::\text{real-normed-vector star}. \text{hnorm } (x + y) \leq \text{hnorm } x + \text{hnorm } y$

*<proof>*

**lemma** *hnorm-triangle-ineq3*:  $\bigwedge x\ y::'a::\text{real-normed-vector star}. |\text{hnorm } x - \text{hnorm } y| \leq \text{hnorm } (x - y)$

*<proof>*

**lemma** *hnorm-scaleR*:  $\bigwedge x::'a::\text{real-normed-vector star}. \text{hnorm } (a *_R x) = |\text{star-of } a| * \text{hnorm } x$

*<proof>*

**lemma** *hnorm-scaleHR*:  $\bigwedge a\ (x::'a::\text{real-normed-vector star}). \text{hnorm } (\text{scaleHR } a\ x)$

$$= |a| * \text{hnorm } x$$

⟨proof⟩

**lemma** *hnorm-mult-ineq*:  $\bigwedge x y :: 'a :: \text{real-normed-algebra star. } \text{hnorm } (x * y) \leq \text{hnorm } x * \text{hnorm } y$

⟨proof⟩

**lemma** *hnorm-mult*:  $\bigwedge x y :: 'a :: \text{real-normed-div-algebra star. } \text{hnorm } (x * y) = \text{hnorm } x * \text{hnorm } y$

⟨proof⟩

**lemma** *hnorm-hyperpow*:  $\bigwedge (x :: 'a :: \{\text{real-normed-div-algebra}\} \text{ star}) n. \text{hnorm } (x \text{ pow } n) = \text{hnorm } x \text{ pow } n$

⟨proof⟩

**lemma** *hnorm-one* [simp]:  $\text{hnorm } (1 :: 'a :: \text{real-normed-div-algebra star}) = 1$

⟨proof⟩

**lemma** *hnorm-zero* [simp]:  $\text{hnorm } (0 :: 'a :: \text{real-normed-vector star}) = 0$

⟨proof⟩

**lemma** *zero-less-hnorm-iff* [simp]:  $\bigwedge x :: 'a :: \text{real-normed-vector star. } 0 < \text{hnorm } x \iff x \neq 0$

⟨proof⟩

**lemma** *hnorm-minus-cancel* [simp]:  $\bigwedge x :: 'a :: \text{real-normed-vector star. } \text{hnorm } (- x) = \text{hnorm } x$

⟨proof⟩

**lemma** *hnorm-minus-commute*:  $\bigwedge a b :: 'a :: \text{real-normed-vector star. } \text{hnorm } (a - b) = \text{hnorm } (b - a)$

⟨proof⟩

**lemma** *hnorm-triangle-ineq2*:  $\bigwedge a b :: 'a :: \text{real-normed-vector star. } \text{hnorm } a - \text{hnorm } b \leq \text{hnorm } (a - b)$

⟨proof⟩

**lemma** *hnorm-triangle-ineq4*:  $\bigwedge a b :: 'a :: \text{real-normed-vector star. } \text{hnorm } (a - b) \leq \text{hnorm } a + \text{hnorm } b$

⟨proof⟩

**lemma** *abs-hnorm-cancel* [simp]:  $\bigwedge a :: 'a :: \text{real-normed-vector star. } |\text{hnorm } a| = \text{hnorm } a$

⟨proof⟩

**lemma** *hnorm-of-hypreal* [simp]:  $\bigwedge r. \text{hnorm } (\text{of-hypreal } r :: 'a :: \text{real-normed-algebra-1 star}) = |r|$

⟨proof⟩

**lemma** *nonzero-hnorm-inverse*:

$\bigwedge a::'a::\text{real-normed-div-algebra star. } a \neq 0 \implies \text{hnorm (inverse a)} = \text{inverse (hnorm a)}$   
 ⟨proof⟩

**lemma** *hnorm-inverse*:

$\bigwedge a::'a::\{\text{real-normed-div-algebra, division-ring}\} \text{ star. } \text{hnorm (inverse a)} = \text{inverse (hnorm a)}$   
 ⟨proof⟩

**lemma** *hnorm-divide*:  $\bigwedge a b::'a::\{\text{real-normed-field, field}\} \text{ star. } \text{hnorm (a / b)} = \text{hnorm a / hnorm b}$

⟨proof⟩

**lemma** *hypreal-hnorm-def [simp]*:  $\bigwedge r::\text{hypreal. } \text{hnorm } r = |r|$

⟨proof⟩

**lemma** *hnorm-add-less*:

$\bigwedge (x::'a::\text{real-normed-vector star}) y r s. \text{hnorm } x < r \implies \text{hnorm } y < s \implies \text{hnorm (x + y)} < r + s$   
 ⟨proof⟩

**lemma** *hnorm-mult-less*:

$\bigwedge (x::'a::\text{real-normed-algebra star}) y r s. \text{hnorm } x < r \implies \text{hnorm } y < s \implies \text{hnorm (x * y)} < r * s$   
 ⟨proof⟩

**lemma** *hnorm-scaleHR-less*:  $|x| < r \implies \text{hnorm } y < s \implies \text{hnorm (scaleHR x y)} < r * s$

⟨proof⟩

## 5.2 Closure Laws for the Standard Reals

**lemma** *Reals-add-cancel*:  $x + y \in \mathbb{R} \implies y \in \mathbb{R} \implies x \in \mathbb{R}$

⟨proof⟩

**lemma** *SReal-hrabs*:  $x \in \mathbb{R} \implies |x| \in \mathbb{R}$

for  $x :: \text{hypreal}$

⟨proof⟩

**lemma** *SReal-hypreal-of-real [simp]*: *hypreal-of-real*  $x \in \mathbb{R}$

⟨proof⟩

**lemma** *SReal-divide-numeral*:  $r \in \mathbb{R} \implies r / (\text{numeral } w::\text{hypreal}) \in \mathbb{R}$

⟨proof⟩

$\varepsilon$  is not in Reals because it is an infinitesimal

**lemma** *SReal-epsilon-not-mem*:  $\varepsilon \notin \mathbb{R}$

⟨proof⟩

**lemma** *SReal-omega-not-mem*:  $\omega \notin \mathbb{R}$

*<proof>*

**lemma** *SReal-UNIV-real*:  $\{x. \text{hypreal-of-real } x \in \mathbb{R}\} = (\text{UNIV}::\text{real set})$

*<proof>*

**lemma** *SReal-iff*:  $x \in \mathbb{R} \longleftrightarrow (\exists y. x = \text{hypreal-of-real } y)$

*<proof>*

**lemma** *hypreal-of-real-image*:  $\text{hypreal-of-real } `(\text{UNIV}::\text{real set}) = \mathbb{R}$

*<proof>*

**lemma** *inv-hypreal-of-real-image*:  $\text{inv hypreal-of-real } ` \mathbb{R} = \text{UNIV}$

*<proof>*

**lemma** *SReal-dense*:  $x \in \mathbb{R} \implies y \in \mathbb{R} \implies x < y \implies \exists r \in \text{Reals. } x < r \wedge r < y$

**for**  $x \ y :: \text{hypreal}$

*<proof>*

### 5.3 Set of Finite Elements is a Subring of the Extended Reals

**lemma** *HFinite-add*:  $x \in \text{HFinite} \implies y \in \text{HFinite} \implies x + y \in \text{HFinite}$

*<proof>*

**lemma** *HFinite-mult*:  $x \in \text{HFinite} \implies y \in \text{HFinite} \implies x * y \in \text{HFinite}$

**for**  $x \ y :: 'a::\text{real-normed-algebra star}$

*<proof>*

**lemma** *HFinite-scaleHR*:  $x \in \text{HFinite} \implies y \in \text{HFinite} \implies \text{scaleHR } x \ y \in \text{HFinite}$

*<proof>*

**lemma** *HFinite-minus-iff*:  $-x \in \text{HFinite} \longleftrightarrow x \in \text{HFinite}$

*<proof>*

**lemma** *HFinite-star-of [simp]*:  $\text{star-of } x \in \text{HFinite}$

*<proof>*

**lemma** *SReal-subset-HFinite*:  $(\mathbb{R}::\text{hypreal set}) \subseteq \text{HFinite}$

*<proof>*

**lemma** *HFiniteD*:  $x \in \text{HFinite} \implies \exists t \in \text{Reals. } \text{hnorm } x < t$

*<proof>*

**lemma** *HFinite-hrabs-iff [iff]*:  $|x| \in \text{HFinite} \longleftrightarrow x \in \text{HFinite}$

**for**  $x :: \text{hypreal}$

*<proof>*

**lemma** *HFinite-hnorm-iff [iff]*:  $\text{hnorm } x \in \text{HFinite} \longleftrightarrow x \in \text{HFinite}$

**for**  $x :: \text{hypreal}$   
 ⟨proof⟩

**lemma** *HFinite-numeral* [simp]: numeral  $w \in \text{HFinite}$   
 ⟨proof⟩

As always with numerals, 0 and 1 are special cases.

**lemma** *HFinite-0* [simp]:  $0 \in \text{HFinite}$   
 ⟨proof⟩

**lemma** *HFinite-1* [simp]:  $1 \in \text{HFinite}$   
 ⟨proof⟩

**lemma** *hrealpow-HFinite*:  $x \in \text{HFinite} \implies x \wedge n \in \text{HFinite}$   
**for**  $x :: 'a :: \{\text{real-normed-algebra}, \text{monoid-mult}\} \text{star}$   
 ⟨proof⟩

**lemma** *HFinite-bounded*:  
**fixes**  $x y :: \text{hypreal}$   
**assumes**  $x \in \text{HFinite}$  **and**  $y: y \leq x$   $0 \leq y$  **shows**  $y \in \text{HFinite}$   
 ⟨proof⟩

#### 5.4 Set of Infinitesimals is a Subring of the Hyperreals

**lemma** *InfinitesimalI*:  $(\bigwedge r. r \in \mathbb{R} \implies 0 < r \implies \text{hnorm } x < r) \implies x \in \text{Infinitesimal}$   
 ⟨proof⟩

**lemma** *InfinitesimalD*:  $x \in \text{Infinitesimal} \implies \forall r \in \text{Reals}. 0 < r \implies \text{hnorm } x < r$   
 ⟨proof⟩

**lemma** *InfinitesimalI2*:  $(\bigwedge r. 0 < r \implies \text{hnorm } x < \text{star-of } r) \implies x \in \text{Infinitesimal}$   
 ⟨proof⟩

**lemma** *InfinitesimalD2*:  $x \in \text{Infinitesimal} \implies 0 < r \implies \text{hnorm } x < \text{star-of } r$   
 ⟨proof⟩

**lemma** *Infinitesimal-zero* [iff]:  $0 \in \text{Infinitesimal}$   
 ⟨proof⟩

**lemma** *Infinitesimal-add*:  
**assumes**  $x \in \text{Infinitesimal}$   $y \in \text{Infinitesimal}$   
**shows**  $x + y \in \text{Infinitesimal}$   
 ⟨proof⟩

**lemma** *Infinitesimal-minus-iff* [simp]:  $-x \in \text{Infinitesimal} \iff x \in \text{Infinitesimal}$   
 ⟨proof⟩

**lemma** *Infinitesimal-hnorm-iff*:  $\text{hnorm } x \in \text{Infinitesimal} \iff x \in \text{Infinitesimal}$

*<proof>*

**lemma** *Infinitesimal-hrabs-iff [iff]:*  $|x| \in \text{Infinitesimal} \longleftrightarrow x \in \text{Infinitesimal}$   
**for**  $x :: \text{hypreal}$   
*<proof>*

**lemma** *Infinitesimal-of-hypreal-iff [simp]:*  
*(of-hypreal*  $x :: 'a :: \text{real-normed-algebra-1 star}$ )  $\in \text{Infinitesimal} \longleftrightarrow x \in \text{Infinitesimal}$   
*<proof>*

**lemma** *Infinitesimal-diff:*  $x \in \text{Infinitesimal} \implies y \in \text{Infinitesimal} \implies x - y \in \text{Infinitesimal}$   
*<proof>*

**lemma** *Infinitesimal-mult:*  
**fixes**  $x y :: 'a :: \text{real-normed-algebra star}$   
**assumes**  $x \in \text{Infinitesimal } y \in \text{Infinitesimal}$   
**shows**  $x * y \in \text{Infinitesimal}$   
*<proof>*

**lemma** *Infinitesimal-HFinite-mult:*  
**fixes**  $x y :: 'a :: \text{real-normed-algebra star}$   
**assumes**  $x \in \text{Infinitesimal } y \in \text{HFinite}$   
**shows**  $x * y \in \text{Infinitesimal}$   
*<proof>*

**lemma** *Infinitesimal-HFinite-scaleHR:*  
**assumes**  $x \in \text{Infinitesimal } y \in \text{HFinite}$   
**shows**  $\text{scaleHR } x y \in \text{Infinitesimal}$   
*<proof>*

**lemma** *Infinitesimal-HFinite-mult2:*  
**fixes**  $x y :: 'a :: \text{real-normed-algebra star}$   
**assumes**  $x \in \text{Infinitesimal } y \in \text{HFinite}$   
**shows**  $y * x \in \text{Infinitesimal}$   
*<proof>*

**lemma** *Infinitesimal-scaleR2:*  
**assumes**  $x \in \text{Infinitesimal}$  **shows**  $a *_R x \in \text{Infinitesimal}$   
*<proof>*

**lemma** *Compl-HFinite:*  $-\text{HFinite} = \text{HInfinite}$   
*<proof>*

**lemma** *HInfinite-inverse-Infinitesimal:*  
 $x \in \text{HInfinite} \implies \text{inverse } x \in \text{Infinitesimal}$   
**for**  $x :: 'a :: \text{real-normed-div-algebra star}$   
*<proof>*

**lemma** *inverse-Infinitesimal-iff-HInfinite*:

$x \neq 0 \implies \text{inverse } x \in \text{Infinitesimal} \longleftrightarrow x \in \text{HInfinite}$

**for**  $x :: 'a::\text{real-normed-div-algebra star}$

*<proof>*

**lemma** *HInfiniteI*:  $(\bigwedge r. r \in \mathbb{R} \implies r < \text{hnorm } x) \implies x \in \text{HInfinite}$

*<proof>*

**lemma** *HInfiniteD*:  $x \in \text{HInfinite} \implies r \in \mathbb{R} \implies r < \text{hnorm } x$

*<proof>*

**lemma** *HInfinite-mult*:

**fixes**  $x y :: 'a::\text{real-normed-div-algebra star}$

**assumes**  $x \in \text{HInfinite } y \in \text{HInfinite}$  **shows**  $x * y \in \text{HInfinite}$

*<proof>*

**lemma** *hypreal-add-zero-less-le-mono*:  $r < x \implies 0 \leq y \implies r < x + y$

**for**  $r x y :: \text{hypreal}$

*<proof>*

**lemma** *HInfinite-add-ge-zero*:  $x \in \text{HInfinite} \implies 0 \leq y \implies 0 \leq x \implies x + y \in \text{HInfinite}$

**for**  $x y :: \text{hypreal}$

*<proof>*

**lemma** *HInfinite-add-ge-zero2*:  $x \in \text{HInfinite} \implies 0 \leq y \implies 0 \leq x \implies y + x \in \text{HInfinite}$

**for**  $x y :: \text{hypreal}$

*<proof>*

**lemma** *HInfinite-add-gt-zero*:  $x \in \text{HInfinite} \implies 0 < y \implies 0 < x \implies x + y \in \text{HInfinite}$

**for**  $x y :: \text{hypreal}$

*<proof>*

**lemma** *HInfinite-minus-iff*:  $-x \in \text{HInfinite} \longleftrightarrow x \in \text{HInfinite}$

*<proof>*

**lemma** *HInfinite-add-le-zero*:  $x \in \text{HInfinite} \implies y \leq 0 \implies x \leq 0 \implies x + y \in \text{HInfinite}$

**for**  $x y :: \text{hypreal}$

*<proof>*

**lemma** *HInfinite-add-lt-zero*:  $x \in \text{HInfinite} \implies y < 0 \implies x < 0 \implies x + y \in \text{HInfinite}$

**for**  $x y :: \text{hypreal}$

*<proof>*

**lemma** *not-Infinitesimal-not-zero*:  $x \notin \text{Infinitesimal} \implies x \neq 0$

*<proof>*

**lemma** *HFfinite-diff-Infinitesimal-hrabs*:

$x \in \text{HFfinite} - \text{Infinitesimal} \implies |x| \in \text{HFfinite} - \text{Infinitesimal}$

**for**  $x :: \text{hypreal}$

*<proof>*

**lemma** *hnorm-le-Infinitesimal*:  $e \in \text{Infinitesimal} \implies \text{hnorm } x \leq e \implies x \in \text{Infinitesimal}$

*<proof>*

**lemma** *hnorm-less-Infinitesimal*:  $e \in \text{Infinitesimal} \implies \text{hnorm } x < e \implies x \in \text{Infinitesimal}$

*<proof>*

**lemma** *hrabs-le-Infinitesimal*:  $e \in \text{Infinitesimal} \implies |x| \leq e \implies x \in \text{Infinitesimal}$

**for**  $x :: \text{hypreal}$

*<proof>*

**lemma** *hrabs-less-Infinitesimal*:  $e \in \text{Infinitesimal} \implies |x| < e \implies x \in \text{Infinitesimal}$

**for**  $x :: \text{hypreal}$

*<proof>*

**lemma** *Infinitesimal-interval*:

$e \in \text{Infinitesimal} \implies e' \in \text{Infinitesimal} \implies e' < x \implies x < e \implies x \in \text{Infinitesimal}$

**for**  $x :: \text{hypreal}$

*<proof>*

**lemma** *Infinitesimal-interval2*:

$e \in \text{Infinitesimal} \implies e' \in \text{Infinitesimal} \implies e' \leq x \implies x \leq e \implies x \in \text{Infinitesimal}$

**for**  $x :: \text{hypreal}$

*<proof>*

**lemma** *lemma-Infinitesimal-hyperpow*:  $x \in \text{Infinitesimal} \implies 0 < N \implies |x \text{ pow } N| \leq |x|$

**for**  $x :: \text{hypreal}$

*<proof>*

**lemma** *Infinitesimal-hyperpow*:  $x \in \text{Infinitesimal} \implies 0 < N \implies x \text{ pow } N \in \text{Infinitesimal}$

**for**  $x :: \text{hypreal}$

*<proof>*

**lemma** *hrealpow-hyperpow-Infinitesimal-iff*:

$(x \hat{\ } n \in \text{Infinitesimal}) \iff x \text{ pow } (\text{hypnat-of-nat } n) \in \text{Infinitesimal}$

*<proof>*

**lemma** *Infinitesimal-hrealpow*:  $x \in \text{Infinitesimal} \implies 0 < n \implies x \hat{\ } n \in \text{Infinitesimal}$



**for**  $x :: \text{hypreal}$   
 ⟨proof⟩

**lemma** *not-Infinitesimal-mult*:

$x \notin \text{Infinitesimal} \implies y \notin \text{Infinitesimal} \implies x * y \notin \text{Infinitesimal}$   
**for**  $x y :: 'a::\text{real-normed-div-algebra star}$   
 ⟨proof⟩

**lemma** *Infinitesimal-mult-disj*:  $x * y \in \text{Infinitesimal} \implies x \in \text{Infinitesimal} \vee y \in \text{Infinitesimal}$

**for**  $x y :: 'a::\text{real-normed-div-algebra star}$   
 ⟨proof⟩

**lemma** *HFinite-Infinitesimal-not-zero*:  $x \in \text{HFinite} - \text{Infinitesimal} \implies x \neq 0$   
 ⟨proof⟩

**lemma** *HFinite-Infinitesimal-diff-mult*:

$x \in \text{HFinite} - \text{Infinitesimal} \implies y \in \text{HFinite} - \text{Infinitesimal} \implies x * y \in \text{HFinite} - \text{Infinitesimal}$   
**for**  $x y :: 'a::\text{real-normed-div-algebra star}$   
 ⟨proof⟩

**lemma** *Infinitesimal-subset-HFinite*:  $\text{Infinitesimal} \subseteq \text{HFinite}$   
 ⟨proof⟩

**lemma** *Infinitesimal-star-of-mult*:  $x \in \text{Infinitesimal} \implies x * \text{star-of } r \in \text{Infinitesimal}$

**for**  $x :: 'a::\text{real-normed-algebra star}$   
 ⟨proof⟩

**lemma** *Infinitesimal-star-of-mult2*:  $x \in \text{Infinitesimal} \implies \text{star-of } r * x \in \text{Infinitesimal}$

**for**  $x :: 'a::\text{real-normed-algebra star}$   
 ⟨proof⟩

## 5.5 The Infinitely Close Relation

**lemma** *mem-infmal-iff*:  $x \in \text{Infinitesimal} \longleftrightarrow x \approx 0$   
 ⟨proof⟩

**lemma** *approx-minus-iff*:  $x \approx y \longleftrightarrow x - y \approx 0$   
 ⟨proof⟩

**lemma** *approx-minus-iff2*:  $x \approx y \longleftrightarrow -y + x \approx 0$   
 ⟨proof⟩

**lemma** *approx-refl [iff]*:  $x \approx x$   
 ⟨proof⟩

**lemma** *approx-sym*:  $x \approx y \implies y \approx x$   
 ⟨proof⟩

**lemma** *approx-trans*:  
 assumes  $x \approx y$   $y \approx z$  shows  $x \approx z$   
 ⟨proof⟩

**lemma** *approx-trans2*:  $r \approx x \implies s \approx x \implies r \approx s$   
 ⟨proof⟩

**lemma** *approx-trans3*:  $x \approx r \implies x \approx s \implies r \approx s$   
 ⟨proof⟩

**lemma** *approx-reorient*:  $x \approx y \longleftrightarrow y \approx x$   
 ⟨proof⟩

Reorientation simplification procedure: reorients (polymorphic)  $0 = x$ ,  $1 = x$ ,  $nnn = x$  provided  $x$  isn't  $0$ ,  $1$  or a numeral.

⟨ML⟩

**lemma** *Infinitesimal-approx-minus*:  $x - y \in \text{Infinitesimal} \longleftrightarrow x \approx y$   
 ⟨proof⟩

**lemma** *approx-monad-iff*:  $x \approx y \longleftrightarrow \text{monad } x = \text{monad } y$   
 ⟨proof⟩

**lemma** *Infinitesimal-approx*:  $x \in \text{Infinitesimal} \implies y \in \text{Infinitesimal} \implies x \approx y$   
 ⟨proof⟩

**lemma** *approx-add*:  $a \approx b \implies c \approx d \implies a + c \approx b + d$   
 ⟨proof⟩

**lemma** *approx-minus*:  $a \approx b \implies -a \approx -b$   
 ⟨proof⟩

**lemma** *approx-minus2*:  $-a \approx -b \implies a \approx b$   
 ⟨proof⟩

**lemma** *approx-minus-cancel [simp]*:  $-a \approx -b \longleftrightarrow a \approx b$   
 ⟨proof⟩

**lemma** *approx-add-minus*:  $a \approx b \implies c \approx d \implies a + -c \approx b + -d$   
 ⟨proof⟩

**lemma** *approx-diff*:  $a \approx b \implies c \approx d \implies a - c \approx b - d$   
 ⟨proof⟩

**lemma** *approx-mult1*:  $a \approx b \implies c \in \text{HFinite} \implies a * c \approx b * c$   
 for  $a b c :: 'a::\text{real-normed-algebra } \text{star}$

*<proof>*

**lemma** *approx-mult2*:  $a \approx b \implies c \in \mathit{HFinite} \implies c * a \approx c * b$   
**for**  $a\ b\ c :: 'a::\mathit{real-normed-algebra\ star}$   
*<proof>*

**lemma** *approx-mult-subst*:  $u \approx v * x \implies x \approx y \implies v \in \mathit{HFinite} \implies u \approx v * y$   
**for**  $u\ v\ x\ y :: 'a::\mathit{real-normed-algebra\ star}$   
*<proof>*

**lemma** *approx-mult-subst2*:  $u \approx x * v \implies x \approx y \implies v \in \mathit{HFinite} \implies u \approx y * v$   
**for**  $u\ v\ x\ y :: 'a::\mathit{real-normed-algebra\ star}$   
*<proof>*

**lemma** *approx-mult-subst-star-of*:  $u \approx x * \mathit{star-of}\ v \implies x \approx y \implies u \approx y * \mathit{star-of}\ v$   
**for**  $u\ x\ y :: 'a::\mathit{real-normed-algebra\ star}$   
*<proof>*

**lemma** *approx-eq-imp*:  $a = b \implies a \approx b$   
*<proof>*

**lemma** *Infinitesimal-minus-approx*:  $x \in \mathit{Infinitesimal} \implies -x \approx x$   
*<proof>*

**lemma** *beX-Infinitesimal-iff*:  $(\exists y \in \mathit{Infinitesimal}. x - z = y) \longleftrightarrow x \approx z$   
*<proof>*

**lemma** *beX-Infinitesimal-iff2*:  $(\exists y \in \mathit{Infinitesimal}. x = z + y) \longleftrightarrow x \approx z$   
*<proof>*

**lemma** *Infinitesimal-add-approx*:  $y \in \mathit{Infinitesimal} \implies x + y = z \implies x \approx z$   
*<proof>*

**lemma** *Infinitesimal-add-approx-self*:  $y \in \mathit{Infinitesimal} \implies x \approx x + y$   
*<proof>*

**lemma** *Infinitesimal-add-approx-self2*:  $y \in \mathit{Infinitesimal} \implies x \approx y + x$   
*<proof>*

**lemma** *Infinitesimal-add-minus-approx-self*:  $y \in \mathit{Infinitesimal} \implies x \approx x + -y$   
*<proof>*

**lemma** *Infinitesimal-add-cancel*:  $y \in \mathit{Infinitesimal} \implies x + y \approx z \implies x \approx z$   
*<proof>*

**lemma** *Infinitesimal-add-right-cancel*:  $y \in \mathit{Infinitesimal} \implies x \approx z + y \implies x \approx z$   
*<proof>*

**lemma** *approx-add-left-cancel*:  $d + b \approx d + c \implies b \approx c$   
 ⟨proof⟩

**lemma** *approx-add-right-cancel*:  $b + d \approx c + d \implies b \approx c$   
 ⟨proof⟩

**lemma** *approx-add-mono1*:  $b \approx c \implies d + b \approx d + c$   
 ⟨proof⟩

**lemma** *approx-add-mono2*:  $b \approx c \implies b + a \approx c + a$   
 ⟨proof⟩

**lemma** *approx-add-left-iff [simp]*:  $a + b \approx a + c \longleftrightarrow b \approx c$   
 ⟨proof⟩

**lemma** *approx-add-right-iff [simp]*:  $b + a \approx c + a \longleftrightarrow b \approx c$   
 ⟨proof⟩

**lemma** *approx-HFinite*:  $x \in \mathit{HFinite} \implies x \approx y \implies y \in \mathit{HFinite}$   
 ⟨proof⟩

**lemma** *approx-star-of-HFinite*:  $x \approx \mathit{star-of} D \implies x \in \mathit{HFinite}$   
 ⟨proof⟩

**lemma** *approx-mult-HFinite*:  $a \approx b \implies c \approx d \implies b \in \mathit{HFinite} \implies d \in \mathit{HFinite} \implies a * c \approx b * d$   
**for**  $a b c d :: 'a::\mathit{real-normed-algebra} \mathit{star}$   
 ⟨proof⟩

**lemma** *scaleHR-left-diff-distrib*:  $\bigwedge a b x. \mathit{scaleHR} (a - b) x = \mathit{scaleHR} a x - \mathit{scaleHR} b x$   
 ⟨proof⟩

**lemma** *approx-scaleR1*:  $a \approx \mathit{star-of} b \implies c \in \mathit{HFinite} \implies \mathit{scaleHR} a c \approx b *_R c$   
 ⟨proof⟩

**lemma** *approx-scaleR2*:  $a \approx b \implies c *_R a \approx c *_R b$   
 ⟨proof⟩

**lemma** *approx-scaleR-HFinite*:  $a \approx \mathit{star-of} b \implies c \approx d \implies d \in \mathit{HFinite} \implies \mathit{scaleHR} a c \approx b *_R d$   
 ⟨proof⟩

**lemma** *approx-mult-star-of*:  $a \approx \mathit{star-of} b \implies c \approx \mathit{star-of} d \implies a * c \approx \mathit{star-of} b * \mathit{star-of} d$   
**for**  $a c :: 'a::\mathit{real-normed-algebra} \mathit{star}$   
 ⟨proof⟩

**lemma** *approx-SReal-mult-cancel-zero*:

**fixes**  $a x :: \text{hypreal}$   
**assumes**  $a \in \mathbb{R} \ a \neq 0 \ a * x \approx 0$  **shows**  $x \approx 0$   
 $\langle \text{proof} \rangle$

**lemma** *approx-mult-SReal1*:  $a \in \mathbb{R} \implies x \approx 0 \implies x * a \approx 0$   
**for**  $a x :: \text{hypreal}$   
 $\langle \text{proof} \rangle$

**lemma** *approx-mult-SReal2*:  $a \in \mathbb{R} \implies x \approx 0 \implies a * x \approx 0$   
**for**  $a x :: \text{hypreal}$   
 $\langle \text{proof} \rangle$

**lemma** *approx-mult-SReal-zero-cancel-iff* [simp]:  $a \in \mathbb{R} \implies a \neq 0 \implies a * x \approx 0 \iff x \approx 0$   
**for**  $a x :: \text{hypreal}$   
 $\langle \text{proof} \rangle$

**lemma** *approx-SReal-mult-cancel*:  
**fixes**  $a w z :: \text{hypreal}$   
**assumes**  $a \in \mathbb{R} \ a \neq 0 \ a * w \approx a * z$  **shows**  $w \approx z$   
 $\langle \text{proof} \rangle$

**lemma** *approx-SReal-mult-cancel-iff1* [simp]:  $a \in \mathbb{R} \implies a \neq 0 \implies a * w \approx a * z \iff w \approx z$   
**for**  $a w z :: \text{hypreal}$   
 $\langle \text{proof} \rangle$

**lemma** *approx-le-bound*:  
**fixes**  $z :: \text{hypreal}$   
**assumes**  $z \leq f \ f \approx g \ g \leq z$  **shows**  $f \approx z$   
 $\langle \text{proof} \rangle$

**lemma** *approx-hnorm*:  $x \approx y \implies \text{hnorm } x \approx \text{hnorm } y$   
**for**  $x y :: 'a::\text{real-normed-vector star}$   
 $\langle \text{proof} \rangle$

## 5.6 Zero is the Only Infinitesimal that is also a Real

**lemma** *Infinitesimal-less-SReal*:  $x \in \mathbb{R} \implies y \in \text{Infinitesimal} \implies 0 < x \implies y < x$   
**for**  $x y :: \text{hypreal}$   
 $\langle \text{proof} \rangle$

**lemma** *Infinitesimal-less-SReal2*:  $y \in \text{Infinitesimal} \implies \forall r \in \text{Reals}. 0 < r \implies y < r$   
**for**  $y :: \text{hypreal}$   
 $\langle \text{proof} \rangle$

**lemma** *SReal-not-Infinitesimal*:  $0 < y \implies y \in \mathbb{R} \implies y \notin \text{Infinitesimal}$

**for**  $y :: \text{hypreal}$   
 ⟨proof⟩

**lemma** *SReal-minus-not-Infinitesimal*:  $y < 0 \implies y \in \mathbb{R} \implies y \notin \text{Infinitesimal}$   
**for**  $y :: \text{hypreal}$   
 ⟨proof⟩

**lemma** *SReal-Int-Infinitesimal-zero*:  $\mathbb{R} \text{ Int } \text{Infinitesimal} = \{0::\text{hypreal}\}$   
 ⟨proof⟩

**lemma** *SReal-Infinitesimal-zero*:  $x \in \mathbb{R} \implies x \in \text{Infinitesimal} \implies x = 0$   
**for**  $x :: \text{hypreal}$   
 ⟨proof⟩

**lemma** *SReal-HFinite-diff-Infinitesimal*:  $x \in \mathbb{R} \implies x \neq 0 \implies x \in \text{HFinite} - \text{Infinitesimal}$   
**for**  $x :: \text{hypreal}$   
 ⟨proof⟩

**lemma** *hypreal-of-real-HFinite-diff-Infinitesimal*:  
 $\text{hypreal-of-real } x \neq 0 \implies \text{hypreal-of-real } x \in \text{HFinite} - \text{Infinitesimal}$   
 ⟨proof⟩

**lemma** *star-of-Infinitesimal-iff-0 [iff]*:  $\text{star-of } x \in \text{Infinitesimal} \longleftrightarrow x = 0$   
 ⟨proof⟩

**lemma** *star-of-HFinite-diff-Infinitesimal*:  $x \neq 0 \implies \text{star-of } x \in \text{HFinite} - \text{Infinitesimal}$   
 ⟨proof⟩

**lemma** *numeral-not-Infinitesimal [simp]*:  
 $\text{numeral } w \neq (0::\text{hypreal}) \implies (\text{numeral } w :: \text{hypreal}) \notin \text{Infinitesimal}$   
 ⟨proof⟩

Again: 1 is a special case, but not 0 this time.

**lemma** *one-not-Infinitesimal [simp]*:  
 $(1::'a::\{\text{real-normed-vector}, \text{zero-neq-one}\} \text{ star}) \notin \text{Infinitesimal}$   
 ⟨proof⟩

**lemma** *approx-SReal-not-zero*:  $y \in \mathbb{R} \implies x \approx y \implies y \neq 0 \implies x \neq 0$   
**for**  $x y :: \text{hypreal}$   
 ⟨proof⟩

**lemma** *HFinite-diff-Infinitesimal-approx*:  
 $x \approx y \implies y \in \text{HFinite} - \text{Infinitesimal} \implies x \in \text{HFinite} - \text{Infinitesimal}$   
 ⟨proof⟩

The premise  $y \neq 0$  is essential; otherwise  $x / y = 0$  and we lose the *HFinite* premise.

**lemma** *Infinitesimal-ratio*:

$y \neq 0 \implies y \in \text{Infinitesimal} \implies x / y \in \text{HFinite} \implies x \in \text{Infinitesimal}$   
**for**  $x y :: 'a::\{\text{real-normed-div-algebra,field}\}$  *star*  
 ⟨proof⟩

**lemma** *Infinitesimal-SReal-divide*:  $x \in \text{Infinitesimal} \implies y \in \mathbb{R} \implies x / y \in \text{Infinitesimal}$

**for**  $x y :: \text{hypreal}$   
 ⟨proof⟩

## 6 Standard Part Theorem

Every finite  $x \in R^*$  is infinitely close to a unique real number (i.e. a member of *Reals*).

### 6.1 Uniqueness: Two Infinitely Close Reals are Equal

**lemma** *star-of-approx-iff* [*simp*]:  $\text{star-of } x \approx \text{star-of } y \iff x = y$   
 ⟨proof⟩

**lemma** *SReal-approx-iff*:  $x \in \mathbb{R} \implies y \in \mathbb{R} \implies x \approx y \iff x = y$   
**for**  $x y :: \text{hypreal}$   
 ⟨proof⟩

**lemma** *numeral-approx-iff* [*simp*]:

$(\text{numeral } v \approx (\text{numeral } w :: 'a::\{\text{numeral,real-normed-vector}\}$  *star*)) =  $(\text{numeral } v = (\text{numeral } w :: 'a))$   
 ⟨proof⟩

And also for  $0 \approx \#nn$  and  $1 \approx \#nn$ ,  $\#nn \approx 0$  and  $\#nn \approx 1$ .

**lemma** [*simp*]:

$(\text{numeral } w \approx (0::'a::\{\text{numeral,real-normed-vector}\}$  *star*)) =  $(\text{numeral } w = (0::'a))$   
 $((0::'a::\{\text{numeral,real-normed-vector}\}$  *star*)  $\approx$   $\text{numeral } w$ ) =  $(\text{numeral } w = (0::'a))$   
 $(\text{numeral } w \approx (1::'b::\{\text{numeral,one,real-normed-vector}\}$  *star*)) =  $(\text{numeral } w = (1::'b))$   
 $((1::'b::\{\text{numeral,one,real-normed-vector}\}$  *star*)  $\approx$   $\text{numeral } w$ ) =  $(\text{numeral } w = (1::'b))$   
 $\neg (0 \approx (1::'c::\{\text{zero-neq-one,real-normed-vector}\}$  *star*))  
 $\neg (1 \approx (0::'c::\{\text{zero-neq-one,real-normed-vector}\}$  *star*))  
 ⟨proof⟩

**lemma** *star-of-approx-numeral-iff* [*simp*]:  $\text{star-of } k \approx \text{numeral } w \iff k = \text{numeral } w$

⟨proof⟩

**lemma** *star-of-approx-zero-iff* [*simp*]:  $\text{star-of } k \approx 0 \iff k = 0$

⟨proof⟩

**lemma** *star-of-approx-one-iff* [simp]: *star-of*  $k \approx 1 \longleftrightarrow k = 1$   
 ⟨proof⟩

**lemma** *approx-unique-real*:  $r \in \mathbb{R} \implies s \in \mathbb{R} \implies r \approx x \implies s \approx x \implies r = s$   
 for  $r\ s :: \text{hypreal}$   
 ⟨proof⟩

## 6.2 Existence of Unique Real Infinitely Close

### 6.2.1 Lifting of the Ub and Lub Properties

**lemma** *hypreal-of-real-isUb-iff*: *isUb*  $\mathbb{R}$  (*hypreal-of-real* ‘  $Q$ ) (*hypreal-of-real*  $Y$ ) =  
*isUb*  $UNIV$   $Q$   $Y$   
 for  $Q :: \text{real set}$  and  $Y :: \text{real}$   
 ⟨proof⟩

**lemma** *hypreal-of-real-isLub-iff*:  
*isLub*  $\mathbb{R}$  (*hypreal-of-real* ‘  $Q$ ) (*hypreal-of-real*  $Y$ ) = *isLub* ( $UNIV :: \text{real set}$ )  $Q$   $Y$   
 (is ?lhs = ?rhs)  
 for  $Q :: \text{real set}$  and  $Y :: \text{real}$   
 ⟨proof⟩

**lemma** *lemma-isUb-hypreal-of-real*: *isUb*  $\mathbb{R}$   $P$   $Y \implies \exists Y_0. \text{isUb } \mathbb{R} P (\text{hypreal-of-real } Y_0)$   
 ⟨proof⟩

**lemma** *lemma-isLub-hypreal-of-real*: *isLub*  $\mathbb{R}$   $P$   $Y \implies \exists Y_0. \text{isLub } \mathbb{R} P (\text{hypreal-of-real } Y_0)$   
 ⟨proof⟩

**lemma** *SReal-complete*:  
 fixes  $P :: \text{hypreal set}$   
 assumes  $\text{isUb } \mathbb{R} P Y P \subseteq \mathbb{R} P \neq \{\}$   
 shows  $\exists t. \text{isLub } \mathbb{R} P t$   
 ⟨proof⟩

Lemmas about lubs.

**lemma** *lemma-st-part-lub*:  
 fixes  $x :: \text{hypreal}$   
 assumes  $x \in \text{HFinite}$   
 shows  $\exists t. \text{isLub } \mathbb{R} \{s. s \in \mathbb{R} \wedge s < x\} t$   
 ⟨proof⟩

**lemma** *hypreal-settle-less-trans*:  $S * <= x \implies x < y \implies S * <= y$   
 for  $x\ y :: \text{hypreal}$   
 ⟨proof⟩

**lemma** *hypreal-gt-isUb*:  $\text{isUb } R S x \implies x < y \implies y \in R \implies \text{isUb } R S y$   
 for  $x\ y :: \text{hypreal}$   
 ⟨proof⟩



**lemma** *lemma-SReal-ub*:  $x \in \mathbb{R} \implies isUb \mathbb{R} \{s. s \in \mathbb{R} \wedge s < x\} x$   
**for**  $x :: hypreal$   
 $\langle proof \rangle$

**lemma** *lemma-SReal-lub*:  
**fixes**  $x :: hypreal$   
**assumes**  $x \in \mathbb{R}$  **shows**  $isLub \mathbb{R} \{s. s \in \mathbb{R} \wedge s < x\} x$   
 $\langle proof \rangle$

**lemma** *lemma-st-part-major*:  
**fixes**  $x r t :: hypreal$   
**assumes**  $x: x \in HFinite$  **and**  $r: r \in \mathbb{R} \ 0 < r$  **and**  $t: isLub \mathbb{R} \{s. s \in \mathbb{R} \wedge s < x\} t$   
**shows**  $|x - t| < r$   
 $\langle proof \rangle$

**lemma** *lemma-st-part-major2*:  
 $x \in HFinite \implies isLub \mathbb{R} \{s. s \in \mathbb{R} \wedge s < x\} t \implies \forall r \in Reals. \ 0 < r \longrightarrow |x - t| < r$   
**for**  $x t :: hypreal$   
 $\langle proof \rangle$

Existence of real and Standard Part Theorem.

**lemma** *lemma-st-part-Ex*:  $x \in HFinite \implies \exists t \in Reals. \forall r \in Reals. \ 0 < r \longrightarrow |x - t| < r$   
**for**  $x :: hypreal$   
 $\langle proof \rangle$

**lemma** *st-part-Ex*:  $x \in HFinite \implies \exists t \in Reals. x \approx t$   
**for**  $x :: hypreal$   
 $\langle proof \rangle$

There is a unique real infinitely close.

**lemma** *st-part-Ex1*:  $x \in HFinite \implies \exists ! t :: hypreal. t \in \mathbb{R} \wedge x \approx t$   
 $\langle proof \rangle$

### 6.3 Finite, Infinite and Infinitesimal

**lemma** *HFinite-Int-HInfinite-empty* [simp]:  $HFinite \text{ Int } HInfinite = \{\}$   
 $\langle proof \rangle$

**lemma** *HFinite-not-HInfinite*:  
**assumes**  $x: x \in HFinite$  **shows**  $x \notin HInfinite$   
 $\langle proof \rangle$

**lemma** *not-HFinite-HInfinite*:  $x \notin HFinite \implies x \in HInfinite$   
 $\langle proof \rangle$

**lemma** *HInfinite-HFinite-disj*:  $x \in HInfinite \vee x \in HFinite$   
 ⟨proof⟩

**lemma** *HInfinite-HFinite-iff*:  $x \in HInfinite \longleftrightarrow x \notin HFinite$   
 ⟨proof⟩

**lemma** *HFinite-HInfinite-iff*:  $x \in HFinite \longleftrightarrow x \notin HInfinite$   
 ⟨proof⟩

**lemma** *HInfinite-diff-HFinite-Infinitesimal-disj*:  
 $x \notin Infinitesimal \implies x \in HInfinite \vee x \in HFinite - Infinitesimal$   
 ⟨proof⟩

**lemma** *HFinite-inverse*:  $x \in HFinite \implies x \notin Infinitesimal \implies \text{inverse } x \in HFinite$   
**for**  $x :: 'a::\text{real-normed-div-algebra star}$   
 ⟨proof⟩

**lemma** *HFinite-inverse2*:  $x \in HFinite - Infinitesimal \implies \text{inverse } x \in HFinite$   
**for**  $x :: 'a::\text{real-normed-div-algebra star}$   
 ⟨proof⟩

Stronger statement possible in fact.

**lemma** *Infinitesimal-inverse-HFinite*:  $x \notin Infinitesimal \implies \text{inverse } x \in HFinite$   
**for**  $x :: 'a::\text{real-normed-div-algebra star}$   
 ⟨proof⟩

**lemma** *HFinite-not-Infinitesimal-inverse*:  
 $x \in HFinite - Infinitesimal \implies \text{inverse } x \in HFinite - Infinitesimal$   
**for**  $x :: 'a::\text{real-normed-div-algebra star}$   
 ⟨proof⟩

**lemma** *approx-inverse*:  
**fixes**  $x y :: 'a::\text{real-normed-div-algebra star}$   
**assumes**  $x \approx y$  **and**  $y: y \in HFinite - Infinitesimal$  **shows**  $\text{inverse } x \approx \text{inverse } y$   
 ⟨proof⟩

**lemmas** *star-of-approx-inverse = star-of-HFinite-diff-Infinitesimal* [THEN [2] *approx-inverse*]

**lemmas** *hypreal-of-real-approx-inverse = hypreal-of-real-HFinite-diff-Infinitesimal* [THEN [2] *approx-inverse*]

**lemma** *inverse-add-Infinitesimal-approx*:  
 $x \in HFinite - Infinitesimal \implies h \in Infinitesimal \implies \text{inverse } (x + h) \approx \text{inverse } x$   
**for**  $x h :: 'a::\text{real-normed-div-algebra star}$   
 ⟨proof⟩

**lemma** *inverse-add-Infinitesimal-approx2*:

$x \in \text{HFinite} - \text{Infinitesimal} \implies h \in \text{Infinitesimal} \implies \text{inverse}(h + x) \approx \text{inverse } x$   
**for**  $x h :: 'a::\text{real-normed-div-algebra star}$   
 ⟨proof⟩

**lemma** *inverse-add-Infinitesimal-approx-Infinitesimal*:

$x \in \text{HFinite} - \text{Infinitesimal} \implies h \in \text{Infinitesimal} \implies \text{inverse}(x + h) - \text{inverse } x \approx h$   
**for**  $x h :: 'a::\text{real-normed-div-algebra star}$   
 ⟨proof⟩

**lemma** *Infinitesimal-square-iff*:  $x \in \text{Infinitesimal} \longleftrightarrow x * x \in \text{Infinitesimal}$

**for**  $x :: 'a::\text{real-normed-div-algebra star}$   
 ⟨proof⟩

**declare** *Infinitesimal-square-iff* [symmetric, simp]

**lemma** *HFinite-square-iff* [simp]:  $x * x \in \text{HFinite} \longleftrightarrow x \in \text{HFinite}$

**for**  $x :: 'a::\text{real-normed-div-algebra star}$   
 ⟨proof⟩

**lemma** *HInfinite-square-iff* [simp]:  $x * x \in \text{HInfinite} \longleftrightarrow x \in \text{HInfinite}$

**for**  $x :: 'a::\text{real-normed-div-algebra star}$   
 ⟨proof⟩

**lemma** *approx-HFinite-mult-cancel*:  $a \in \text{HFinite} - \text{Infinitesimal} \implies a * w \approx a * z \implies w \approx z$

**for**  $a w z :: 'a::\text{real-normed-div-algebra star}$   
 ⟨proof⟩

**lemma** *approx-HFinite-mult-cancel-iff1*:  $a \in \text{HFinite} - \text{Infinitesimal} \implies a * w \approx a * z \longleftrightarrow w \approx z$

**for**  $a w z :: 'a::\text{real-normed-div-algebra star}$   
 ⟨proof⟩

**lemma** *HInfinite-HFinite-add-cancel*:  $x + y \in \text{HInfinite} \implies y \in \text{HFinite} \implies x \in \text{HInfinite}$

⟨proof⟩

**lemma** *HInfinite-HFinite-add*:  $x \in \text{HInfinite} \implies y \in \text{HFinite} \implies x + y \in \text{HInfinite}$

⟨proof⟩

**lemma** *HInfinite-ge-HInfinite*:  $x \in \text{HInfinite} \implies x \leq y \implies 0 \leq x \implies y \in \text{HInfinite}$

**for**  $x y :: \text{hypreal}$   
 ⟨proof⟩

**lemma** *Infinitesimal-inverse-HInfinite*:  $x \in \text{Infinitesimal} \implies x \neq 0 \implies \text{inverse } x \in \text{HInfinite}$

**for**  $x :: 'a::\text{real-normed-div-algebra star}$   
 $\langle \text{proof} \rangle$

**lemma** *HInfinite-HFinite-not-Infinitesimal-mult*:

$x \in \text{HInfinite} \implies y \in \text{HFinite} - \text{Infinitesimal} \implies x * y \in \text{HInfinite}$   
**for**  $x y :: 'a::\text{real-normed-div-algebra star}$   
 $\langle \text{proof} \rangle$

**lemma** *HInfinite-HFinite-not-Infinitesimal-mult2*:

$x \in \text{HInfinite} \implies y \in \text{HFinite} - \text{Infinitesimal} \implies y * x \in \text{HInfinite}$   
**for**  $x y :: 'a::\text{real-normed-div-algebra star}$   
 $\langle \text{proof} \rangle$

**lemma** *HInfinite-gt-SReal*:  $x \in \text{HInfinite} \implies 0 < x \implies y \in \mathbb{R} \implies y < x$

**for**  $x y :: \text{hypreal}$   
 $\langle \text{proof} \rangle$

**lemma** *HInfinite-gt-zero-gt-one*:  $x \in \text{HInfinite} \implies 0 < x \implies 1 < x$

**for**  $x :: \text{hypreal}$   
 $\langle \text{proof} \rangle$

**lemma** *not-HInfinite-one [simp]*:  $1 \notin \text{HInfinite}$

$\langle \text{proof} \rangle$

**lemma** *approx-hrabs-disj*:  $|x| \approx x \vee |x| \approx -x$

**for**  $x :: \text{hypreal}$   
 $\langle \text{proof} \rangle$

## 6.4 Theorems about Monads

**lemma** *monad-hrabs-Un-subset*:  $\text{monad } |x| \leq \text{monad } x \cup \text{monad } (-x)$

**for**  $x :: \text{hypreal}$   
 $\langle \text{proof} \rangle$

**lemma** *Infinitesimal-monad-eq*:  $e \in \text{Infinitesimal} \implies \text{monad } (x + e) = \text{monad } x$

$\langle \text{proof} \rangle$

**lemma** *mem-monad-iff*:  $u \in \text{monad } x \longleftrightarrow -u \in \text{monad } (-x)$

$\langle \text{proof} \rangle$

**lemma** *Infinitesimal-monad-zero-iff*:  $x \in \text{Infinitesimal} \longleftrightarrow x \in \text{monad } 0$

$\langle \text{proof} \rangle$

**lemma** *monad-zero-minus-iff*:  $x \in \text{monad } 0 \longleftrightarrow -x \in \text{monad } 0$

$\langle \text{proof} \rangle$

**lemma** *monad-zero-hrabs-iff*:  $x \in \text{monad } 0 \longleftrightarrow |x| \in \text{monad } 0$

**for**  $x :: \text{hypreal}$   
 ⟨proof⟩

**lemma** *mem-monad-self* [simp]:  $x \in \text{monad } x$   
 ⟨proof⟩

## 6.5 Proof that $x \approx y$ implies $|x| \approx |y|$

**lemma** *approx-subset-monad*:  $x \approx y \implies \{x, y\} \leq \text{monad } x$   
 ⟨proof⟩

**lemma** *approx-subset-monad2*:  $x \approx y \implies \{x, y\} \leq \text{monad } y$   
 ⟨proof⟩

**lemma** *mem-monad-approx*:  $u \in \text{monad } x \implies x \approx u$   
 ⟨proof⟩

**lemma** *approx-mem-monad*:  $x \approx u \implies u \in \text{monad } x$   
 ⟨proof⟩

**lemma** *approx-mem-monad2*:  $x \approx u \implies x \in \text{monad } u$   
 ⟨proof⟩

**lemma** *approx-mem-monad-zero*:  $x \approx y \implies x \in \text{monad } 0 \implies y \in \text{monad } 0$   
 ⟨proof⟩

**lemma** *Infinitesimal-approx-hrabs*:  $x \approx y \implies x \in \text{Infinitesimal} \implies |x| \approx |y|$   
**for**  $x y :: \text{hypreal}$   
 ⟨proof⟩

**lemma** *less-Infinitesimal-less*:  $0 < x \implies x \notin \text{Infinitesimal} \implies e \in \text{Infinitesimal} \implies e < x$   
**for**  $x :: \text{hypreal}$   
 ⟨proof⟩

**lemma** *Ball-mem-monad-gt-zero*:  $0 < x \implies x \notin \text{Infinitesimal} \implies u \in \text{monad } x \implies 0 < u$   
**for**  $u x :: \text{hypreal}$   
 ⟨proof⟩

**lemma** *Ball-mem-monad-less-zero*:  $x < 0 \implies x \notin \text{Infinitesimal} \implies u \in \text{monad } x \implies u < 0$   
**for**  $u x :: \text{hypreal}$   
 ⟨proof⟩

**lemma** *lemma-approx-gt-zero*:  $0 < x \implies x \notin \text{Infinitesimal} \implies x \approx y \implies 0 < y$   
**for**  $x y :: \text{hypreal}$   
 ⟨proof⟩

**lemma** *lemma-approx-less-zero*:  $x < 0 \implies x \notin \text{Infinitesimal} \implies x \approx y \implies y < 0$

**for**  $x\ y :: \text{hypreal}$   
 ⟨proof⟩

**lemma** *approx-hrabs*:  $x \approx y \implies |x| \approx |y|$

**for**  $x\ y :: \text{hypreal}$   
 ⟨proof⟩

**lemma** *approx-hrabs-zero-cancel*:  $|x| \approx 0 \implies x \approx 0$

**for**  $x :: \text{hypreal}$   
 ⟨proof⟩

**lemma** *approx-hrabs-add-Infinitesimal*:  $e \in \text{Infinitesimal} \implies |x| \approx |x + e|$

**for**  $e\ x :: \text{hypreal}$   
 ⟨proof⟩

**lemma** *approx-hrabs-add-minus-Infinitesimal*:  $e \in \text{Infinitesimal} \implies |x| \approx |x + -e|$

**for**  $e\ x :: \text{hypreal}$   
 ⟨proof⟩

**lemma** *hrabs-add-Infinitesimal-cancel*:

$e \in \text{Infinitesimal} \implies e' \in \text{Infinitesimal} \implies |x + e| = |y + e'| \implies |x| \approx |y|$   
**for**  $e\ e'\ x\ y :: \text{hypreal}$   
 ⟨proof⟩

**lemma** *hrabs-add-minus-Infinitesimal-cancel*:

$e \in \text{Infinitesimal} \implies e' \in \text{Infinitesimal} \implies |x + -e| = |y + -e'| \implies |x| \approx |y|$   
**for**  $e\ e'\ x\ y :: \text{hypreal}$   
 ⟨proof⟩

## 6.6 More HFinite and Infinitesimal Theorems

Interesting slightly counterintuitive theorem: necessary for proving that an open interval is an NS open set.

**lemma** *Infinitesimal-add-hypreal-of-real-less*:

**assumes**  $x < y$  **and**  $u: u \in \text{Infinitesimal}$   
**shows**  $\text{hypreal-of-real } x + u < \text{hypreal-of-real } y$   
 ⟨proof⟩

**lemma** *Infinitesimal-add-hrabs-hypreal-of-real-less*:

$x \in \text{Infinitesimal} \implies |\text{hypreal-of-real } r| < \text{hypreal-of-real } y \implies$   
 $|\text{hypreal-of-real } r + x| < \text{hypreal-of-real } y$   
 ⟨proof⟩

**lemma** *Infinitesimal-add-hrabs-hypreal-of-real-less2*:

$x \in \text{Infinitesimal} \implies |\text{hypreal-of-real } r| < \text{hypreal-of-real } y \implies$   
 $|x + \text{hypreal-of-real } r| < \text{hypreal-of-real } y$

*<proof>*

**lemma** *hypreal-of-real-le-add-Infininitesimal-cancel:*

**assumes** *le: hypreal-of-real  $x + u \leq$  hypreal-of-real  $y + v$*

**and** *u:  $u \in$  Infinitesimal and v:  $v \in$  Infinitesimal*

**shows** *hypreal-of-real  $x \leq$  hypreal-of-real  $y$*

*<proof>*

**lemma** *hypreal-of-real-le-add-Infininitesimal-cancel2:*

*$u \in$  Infinitesimal  $\implies v \in$  Infinitesimal  $\implies$*

*hypreal-of-real  $x + u \leq$  hypreal-of-real  $y + v \implies x \leq y$*

*<proof>*

**lemma** *hypreal-of-real-less-Infininitesimal-le-zero:*

*hypreal-of-real  $x < e \implies e \in$  Infinitesimal  $\implies$  hypreal-of-real  $x \leq 0$*

*<proof>*

**lemma** *Infinitesimal-add-not-zero:  $h \in$  Infinitesimal  $\implies x \neq 0 \implies$  star-of  $x + h \neq 0$*

*<proof>*

**lemma** *monad-hrabs-less:  $y \in$  monad  $x \implies 0 < hypreal-of-real e \implies |y - x| < hypreal-of-real e$*

*<proof>*

**lemma** *mem-monad-SReal-HFfinite:  $x \in$  monad (hypreal-of-real  $a$ )  $\implies x \in$  HFfinite*

*<proof>*

## 6.7 Theorems about Standard Part

**lemma** *st-approx-self:  $x \in$  HFfinite  $\implies st x \approx x$*

*<proof>*

**lemma** *st-SReal:  $x \in$  HFfinite  $\implies st x \in \mathbf{R}$*

*<proof>*

**lemma** *st-HFfinite:  $x \in$  HFfinite  $\implies st x \in$  HFfinite*

*<proof>*

**lemma** *st-unique:  $r \in \mathbf{R} \implies r \approx x \implies st x = r$*

*<proof>*

**lemma** *st-SReal-eq:  $x \in \mathbf{R} \implies st x = x$*

*<proof>*

**lemma** *st-hypreal-of-real [simp]:  $st$  (hypreal-of-real  $x$ ) = hypreal-of-real  $x$*

*<proof>*

**lemma** *st-eq-approx*:  $x \in HFinite \implies y \in HFinite \implies st\ x = st\ y \implies x \approx y$   
 ⟨proof⟩

**lemma** *approx-st-eq*:  
 assumes  $x: x \in HFinite$  and  $y: y \in HFinite$  and  $xy: x \approx y$   
 shows  $st\ x = st\ y$   
 ⟨proof⟩

**lemma** *st-eq-approx-iff*:  $x \in HFinite \implies y \in HFinite \implies x \approx y \iff st\ x = st\ y$   
 ⟨proof⟩

**lemma** *st-Infinitesimal-add-SReal*:  $x \in \mathbb{R} \implies e \in Infinitesimal \implies st\ (x + e) = x$   
 ⟨proof⟩

**lemma** *st-Infinitesimal-add-SReal2*:  $x \in \mathbb{R} \implies e \in Infinitesimal \implies st\ (e + x) = x$   
 ⟨proof⟩

**lemma** *HFinite-st-Infinitesimal-add*:  $x \in HFinite \implies \exists e \in Infinitesimal. x = st(x) + e$   
 ⟨proof⟩

**lemma** *st-add*:  $x \in HFinite \implies y \in HFinite \implies st\ (x + y) = st\ x + st\ y$   
 ⟨proof⟩

**lemma** *st-numeral [simp]*:  $st\ (numeral\ w) = numeral\ w$   
 ⟨proof⟩

**lemma** *st-neg-numeral [simp]*:  $st\ (-\ numeral\ w) = -\ numeral\ w$   
 ⟨proof⟩

**lemma** *st-0 [simp]*:  $st\ 0 = 0$   
 ⟨proof⟩

**lemma** *st-1 [simp]*:  $st\ 1 = 1$   
 ⟨proof⟩

**lemma** *st-neg-1 [simp]*:  $st\ (-\ 1) = -\ 1$   
 ⟨proof⟩

**lemma** *st-minus*:  $x \in HFinite \implies st\ (-\ x) = -\ st\ x$   
 ⟨proof⟩

**lemma** *st-diff*:  $\llbracket x \in HFinite; y \in HFinite \rrbracket \implies st\ (x - y) = st\ x - st\ y$   
 ⟨proof⟩

**lemma** *st-mult*:  $\llbracket x \in HFinite; y \in HFinite \rrbracket \implies st\ (x * y) = st\ x * st\ y$   
 ⟨proof⟩



**lemma** *st-Infinitesimal*:  $x \in \text{Infinitesimal} \implies \text{st } x = 0$   
 ⟨proof⟩

**lemma** *st-not-Infinitesimal*:  $\text{st}(x) \neq 0 \implies x \notin \text{Infinitesimal}$   
 ⟨proof⟩

**lemma** *st-inverse*:  $x \in \text{HFinite} \implies \text{st } x \neq 0 \implies \text{st } (\text{inverse } x) = \text{inverse } (\text{st } x)$   
 ⟨proof⟩

**lemma** *st-divide [simp]*:  $x \in \text{HFinite} \implies y \in \text{HFinite} \implies \text{st } y \neq 0 \implies \text{st } (x / y) = \text{st } x / \text{st } y$   
 ⟨proof⟩

**lemma** *st-idempotent [simp]*:  $x \in \text{HFinite} \implies \text{st } (\text{st } x) = \text{st } x$   
 ⟨proof⟩

**lemma** *Infinitesimal-add-st-less*:  
 $x \in \text{HFinite} \implies y \in \text{HFinite} \implies u \in \text{Infinitesimal} \implies \text{st } x < \text{st } y \implies \text{st } x + u < \text{st } y$   
 ⟨proof⟩

**lemma** *Infinitesimal-add-st-le-cancel*:  
 $x \in \text{HFinite} \implies y \in \text{HFinite} \implies u \in \text{Infinitesimal} \implies \text{st } x \leq \text{st } y + u \implies \text{st } x \leq \text{st } y$   
 ⟨proof⟩

**lemma** *st-le*:  $x \in \text{HFinite} \implies y \in \text{HFinite} \implies x \leq y \implies \text{st } x \leq \text{st } y$   
 ⟨proof⟩

**lemma** *st-zero-le*:  $0 \leq x \implies x \in \text{HFinite} \implies 0 \leq \text{st } x$   
 ⟨proof⟩

**lemma** *st-zero-ge*:  $x \leq 0 \implies x \in \text{HFinite} \implies \text{st } x \leq 0$   
 ⟨proof⟩

**lemma** *st-hrabs*:  $x \in \text{HFinite} \implies |\text{st } x| = \text{st } |x|$   
 ⟨proof⟩

## 6.8 Alternative Definitions using Free Ultrafilter

### 6.8.1 *HFinite*

**lemma** *HFinite-FreeUltrafilterNat*:  
 assumes  $\text{star-}n \ X \in \text{HFinite}$   
 shows  $\exists u. \text{eventually } (\lambda n. \text{norm } (X \ n) < u) \ \mathcal{U}$   
 ⟨proof⟩

**lemma** *FreeUltrafilterNat-HFinite*:  
 assumes  $\text{eventually } (\lambda n. \text{norm } (X \ n) < u) \ \mathcal{U}$

**shows**  $star\text{-}n\ X \in HFinite$   
 ⟨proof⟩

**lemma** *HFinite-FreeUltrafilterNat-iff*:  
 $star\text{-}n\ X \in HFinite \longleftrightarrow (\exists u. eventually\ (\lambda n. norm\ (X\ n) < u)\ \mathcal{U})$   
 ⟨proof⟩

### 6.8.2 *HInfinite*

Exclude this type of sets from free ultrafilter for Infinite numbers!

**lemma** *FreeUltrafilterNat-const-Finite*:  
 $eventually\ (\lambda n. norm\ (X\ n) = u)\ \mathcal{U} \implies star\text{-}n\ X \in HFinite$   
 ⟨proof⟩

**lemma** *HInfinite-FreeUltrafilterNat*:  
**assumes**  $star\text{-}n\ X \in HInfinite$  **shows**  $\forall_F\ n\ in\ \mathcal{U}. u < norm\ (X\ n)$   
 ⟨proof⟩

**lemma** *FreeUltrafilterNat-HInfinite*:  
**assumes**  $\bigwedge u. eventually\ (\lambda n. u < norm\ (X\ n))\ \mathcal{U}$   
**shows**  $star\text{-}n\ X \in HInfinite$   
 ⟨proof⟩

**lemma** *HInfinite-FreeUltrafilterNat-iff*:  
 $star\text{-}n\ X \in HInfinite \longleftrightarrow (\forall u. eventually\ (\lambda n. u < norm\ (X\ n))\ \mathcal{U})$   
 ⟨proof⟩

### 6.8.3 *Infinitesimal*

**lemma** *ball-SReal-eq*:  $(\forall x::hypreal \in Reals. P\ x) \longleftrightarrow (\forall x::real. P\ (star\text{-}of\ x))$   
 ⟨proof⟩

**lemma** *Infinitesimal-FreeUltrafilterNat-iff*:  
 $(star\text{-}n\ X \in Infinitesimal) = (\forall u > 0. eventually\ (\lambda n. norm\ (X\ n) < u)\ \mathcal{U})$  **(is**  
 $?lhs = ?rhs)$   
 ⟨proof⟩

Infinitesimals as smaller than  $1/n$  for all  $n::nat (> 0)$ .

**lemma** *lemma-Infinitesimal*:  $(\forall r. 0 < r \longrightarrow x < r) \longleftrightarrow (\forall n. x < inverse\ (real\ (Suc\ n)))$   
 ⟨proof⟩

**lemma** *lemma-Infinitesimal2*:  
 $(\forall r \in Reals. 0 < r \longrightarrow x < r) \longleftrightarrow (\forall n. x < inverse(hypreal\text{-}of\text{-}nat\ (Suc\ n)))$   
**(is - = ?rhs)**  
 ⟨proof⟩

**lemma** *Infinitesimal-hypreal-of-nat-iff:*

$$\text{Infinitesimal} = \{x. \forall n. \text{hnorm } x < \text{inverse } (\text{hypreal-of-nat } (\text{Suc } n))\}$$

*<proof>*

## 6.9 Proof that $\omega$ is an infinite number

It will follow that  $\varepsilon$  is an infinitesimal number.

**lemma** *Suc-Un-eq:*  $\{n. n < \text{Suc } m\} = \{n. n < m\} \cup \{n. n = m\}$

*<proof>*

Prove that any segment is finite and hence cannot belong to  $\mathcal{U}$ .

**lemma** *finite-real-of-nat-segment:*  $\text{finite } \{n::\text{nat}. \text{real } n < \text{real } (m::\text{nat})\}$

*<proof>*

**lemma** *finite-real-of-nat-less-real:*  $\text{finite } \{n::\text{nat}. \text{real } n < u\}$

*<proof>*

**lemma** *finite-real-of-nat-le-real:*  $\text{finite } \{n::\text{nat}. \text{real } n \leq u\}$

*<proof>*

**lemma** *finite-rabs-real-of-nat-le-real:*  $\text{finite } \{n::\text{nat}. |\text{real } n| \leq u\}$

*<proof>*

**lemma** *rabs-real-of-nat-le-real-FreeUltrafilterNat:*

$$\neg \text{eventually } (\lambda n. |\text{real } n| \leq u) \mathcal{U}$$

*<proof>*

**lemma** *FreeUltrafilterNat-nat-gt-real:*  $\text{eventually } (\lambda n. u < \text{real } n) \mathcal{U}$

*<proof>*

The complement of  $\{n. |\text{real } n| \leq u\} = \{n. u < |\text{real } n|\}$  is in  $\mathcal{U}$  by property of (free) ultrafilters.

$\omega$  is a member of *HInfinite*.

**theorem** *HInfinite-omega [simp]:*  $\omega \in \text{HInfinite}$

*<proof>*

Epsilon is a member of *Infinitesimal*.

**lemma** *Infinitesimal-epsilon [simp]:*  $\varepsilon \in \text{Infinitesimal}$

*<proof>*

**lemma** *HFinite-epsilon [simp]:*  $\varepsilon \in \text{HFinite}$

*<proof>*

**lemma** *epsilon-approx-zero [simp]:*  $\varepsilon \approx 0$

*<proof>*

Needed for proof that we define a hyperreal  $[\langle X(n) \rangle] \approx \text{hypreal-of-real } a$  given that  $\forall n. |X n - a| < 1/n$ . Used in proof of *NSLIM*  $\Rightarrow$  *LIM*.

**lemma** *real-of-nat-less-inverse-iff*:  $0 < u \implies u < \text{inverse}(\text{real}(\text{Suc } n)) \longleftrightarrow \text{real}(\text{Suc } n) < \text{inverse } u$   
 ⟨proof⟩

**lemma** *finite-inverse-real-of-posnat-gt-real*:  $0 < u \implies \text{finite } \{n. u < \text{inverse}(\text{real}(\text{Suc } n))\}$   
 ⟨proof⟩

**lemma** *finite-inverse-real-of-posnat-ge-real*:  
 assumes  $0 < u$   
 shows  $\text{finite } \{n. u \leq \text{inverse}(\text{real}(\text{Suc } n))\}$   
 ⟨proof⟩

**lemma** *inverse-real-of-posnat-ge-real-FreeUltrafilterNat*:  
 $0 < u \implies \neg \text{eventually } (\lambda n. u \leq \text{inverse}(\text{real}(\text{Suc } n))) \mathcal{U}$   
 ⟨proof⟩

**lemma** *FreeUltrafilterNat-inverse-real-of-posnat*:  
 $0 < u \implies \text{eventually } (\lambda n. \text{inverse}(\text{real}(\text{Suc } n)) < u) \mathcal{U}$   
 ⟨proof⟩

Example of an hypersequence (i.e. an extended standard sequence) whose term with an hypernatural suffix is an infinitesimal i.e. the whn’nth term of the hypersequence is a member of Infinitesimal

**lemma** *SEQ-Infinitesimal*:  $( *f* (\lambda n::\text{nat}. \text{inverse}(\text{real}(\text{Suc } n)))) \text{whn} \in \text{Infinitesimal}$   
 ⟨proof⟩

Example where we get a hyperreal from a real sequence for which a particular property holds. The theorem is used in proofs about equivalence of nonstandard and standard neighbourhoods. Also used for equivalence of nonstandard and standard definitions of pointwise limit.

$|X(n) - x| < 1/n \implies [\langle X \ n \rangle] - \text{hypreal-of-real } x \in \text{Infinitesimal}$

**lemma** *real-seq-to-hypreal-Infinitesimal*:  
 $\forall n. \text{norm } (X \ n - x) < \text{inverse}(\text{real}(\text{Suc } n)) \implies \text{star-n } X - \text{star-of } x \in \text{Infinitesimal}$   
 ⟨proof⟩

**lemma** *real-seq-to-hypreal-approx*:  
 $\forall n. \text{norm } (X \ n - x) < \text{inverse}(\text{real}(\text{Suc } n)) \implies \text{star-n } X \approx \text{star-of } x$   
 ⟨proof⟩

**lemma** *real-seq-to-hypreal-approx2*:  
 $\forall n. \text{norm } (x - X \ n) < \text{inverse}(\text{real}(\text{Suc } n)) \implies \text{star-n } X \approx \text{star-of } x$   
 ⟨proof⟩

**lemma** *real-seq-to-hypreal-Infinitesimal2*:

$\forall n. \text{norm}(X \ n - Y \ n) < \text{inverse}(\text{real}(\text{Suc } n)) \implies \text{star-}n \ X - \text{star-}n \ Y \in \text{Infinitesimal}$   
 {proof}

end

## 7 Nonstandard Complex Numbers

theory NSComplex  
 imports NSA  
 begin

type-synonym hcomplex = complex star

abbreviation hcomplex-of-complex :: complex  $\Rightarrow$  complex star  
 where hcomplex-of-complex  $\equiv$  star-of

abbreviation hcmol :: complex star  $\Rightarrow$  real star  
 where hcmol  $\equiv$  hnorm

### 7.0.1 Real and Imaginary parts

definition hRe :: hcomplex  $\Rightarrow$  hypreal  
 where hRe =  $*f*$  Re

definition hIm :: hcomplex  $\Rightarrow$  hypreal  
 where hIm =  $*f*$  Im

### 7.0.2 Imaginary unit

definition iii :: hcomplex  
 where iii = star-of i

### 7.0.3 Complex conjugate

definition hcnj :: hcomplex  $\Rightarrow$  hcomplex  
 where hcnj =  $*f*$  cnj

### 7.0.4 Argand

definition hsgn :: hcomplex  $\Rightarrow$  hcomplex  
 where hsgn =  $*f*$  sgn

definition harg :: hcomplex  $\Rightarrow$  hypreal  
 where harg =  $*f*$  Arg

definition — abbreviation for  $\cos a + i \sin a$   
 hcis :: hypreal  $\Rightarrow$  hcomplex  
 where hcis =  $*f*$  cis

**7.0.5 Injection from hyperreals**

**abbreviation**  $hcomplex\text{-of-hypreal} :: \text{hypreal} \Rightarrow hcomplex$   
**where**  $hcomplex\text{-of-hypreal} \equiv \text{of-hypreal}$

**definition** — abbreviation for  $r * (\cos a + i \sin a)$   
 $hrcis :: \text{hypreal} \Rightarrow \text{hypreal} \Rightarrow hcomplex$   
**where**  $hrcis = *f2* rcis$

**7.0.6  $e^{\wedge}(x + iy)$** 

**definition**  $hExp :: hcomplex \Rightarrow hcomplex$   
**where**  $hExp = *f* exp$

**definition**  $HComplex :: \text{hypreal} \Rightarrow \text{hypreal} \Rightarrow hcomplex$   
**where**  $HComplex = *f2* Complex$

**lemmas**  $hcomplex\text{-defs} [\text{transfer-unfold}] =$   
 $hRe\text{-def } hIm\text{-def } iii\text{-def } hcnj\text{-def } hsgn\text{-def } harg\text{-def } hcis\text{-def}$   
 $hrcis\text{-def } hExp\text{-def } HComplex\text{-def}$

**lemma**  $Standard\text{-}hRe$   $[\text{simp}]: x \in Standard \Longrightarrow hRe x \in Standard$   
 $\langle \text{proof} \rangle$

**lemma**  $Standard\text{-}hIm$   $[\text{simp}]: x \in Standard \Longrightarrow hIm x \in Standard$   
 $\langle \text{proof} \rangle$

**lemma**  $Standard\text{-}iii$   $[\text{simp}]: iii \in Standard$   
 $\langle \text{proof} \rangle$

**lemma**  $Standard\text{-}hcnj$   $[\text{simp}]: x \in Standard \Longrightarrow hcnj x \in Standard$   
 $\langle \text{proof} \rangle$

**lemma**  $Standard\text{-}hsgn$   $[\text{simp}]: x \in Standard \Longrightarrow hsgn x \in Standard$   
 $\langle \text{proof} \rangle$

**lemma**  $Standard\text{-}harg$   $[\text{simp}]: x \in Standard \Longrightarrow harg x \in Standard$   
 $\langle \text{proof} \rangle$

**lemma**  $Standard\text{-}hcis$   $[\text{simp}]: r \in Standard \Longrightarrow hcis r \in Standard$   
 $\langle \text{proof} \rangle$

**lemma**  $Standard\text{-}hExp$   $[\text{simp}]: x \in Standard \Longrightarrow hExp x \in Standard$   
 $\langle \text{proof} \rangle$

**lemma**  $Standard\text{-}hrcis$   $[\text{simp}]: r \in Standard \Longrightarrow s \in Standard \Longrightarrow hrcis r s \in Standard$   
 $\langle \text{proof} \rangle$

**lemma**  $Standard\text{-}HComplex$   $[\text{simp}]: r \in Standard \Longrightarrow s \in Standard \Longrightarrow HComplex$

$r\ s \in \text{Standard}$   
 ⟨proof⟩

**lemma** *hcmmod-def*:  $hcmmod = *f* cmod$   
 ⟨proof⟩

## 7.1 Properties of Nonstandard Real and Imaginary Parts

**lemma** *hcomplex-hRe-hIm-cancel-iff*:  $\bigwedge w\ z. w = z \longleftrightarrow hRe\ w = hRe\ z \wedge hIm\ w = hIm\ z$   
 ⟨proof⟩

**lemma** *hcomplex-equality* [*intro?*]:  $\bigwedge z\ w. hRe\ z = hRe\ w \implies hIm\ z = hIm\ w \implies z = w$   
 ⟨proof⟩

**lemma** *hcomplex-hRe-zero* [*simp*]:  $hRe\ 0 = 0$   
 ⟨proof⟩

**lemma** *hcomplex-hIm-zero* [*simp*]:  $hIm\ 0 = 0$   
 ⟨proof⟩

**lemma** *hcomplex-hRe-one* [*simp*]:  $hRe\ 1 = 1$   
 ⟨proof⟩

**lemma** *hcomplex-hIm-one* [*simp*]:  $hIm\ 1 = 0$   
 ⟨proof⟩

## 7.2 Addition for Nonstandard Complex Numbers

**lemma** *hRe-add*:  $\bigwedge x\ y. hRe\ (x + y) = hRe\ x + hRe\ y$   
 ⟨proof⟩

**lemma** *hIm-add*:  $\bigwedge x\ y. hIm\ (x + y) = hIm\ x + hIm\ y$   
 ⟨proof⟩

## 7.3 More Minus Laws

**lemma** *hRe-minus*:  $\bigwedge z. hRe\ (-z) = -hRe\ z$   
 ⟨proof⟩

**lemma** *hIm-minus*:  $\bigwedge z. hIm\ (-z) = -hIm\ z$   
 ⟨proof⟩

**lemma** *hcomplex-add-minus-eq-minus*:  $x + y = 0 \implies x = -y$   
 for  $x\ y :: \text{hcomplex}$   
 ⟨proof⟩

**lemma** *hcomplex-i-mult-eq* [*simp*]:  $iii * iii = -1$   
 ⟨proof⟩

**lemma** *hcomplex-i-mult-left* [*simp*]:  $\bigwedge z. \text{iii} * (\text{iii} * z) = - z$   
 ⟨*proof*⟩

**lemma** *hcomplex-i-not-zero* [*simp*]:  $\text{iii} \neq 0$   
 ⟨*proof*⟩

## 7.4 More Multiplication Laws

**lemma** *hcomplex-mult-minus-one*:  $- 1 * z = - z$   
**for**  $z :: \text{hcomplex}$   
 ⟨*proof*⟩

**lemma** *hcomplex-mult-minus-one-right*:  $z * - 1 = - z$   
**for**  $z :: \text{hcomplex}$   
 ⟨*proof*⟩

**lemma** *hcomplex-mult-left-cancel*:  $c \neq 0 \implies c * a = c * b \iff a = b$   
**for**  $a b c :: \text{hcomplex}$   
 ⟨*proof*⟩

**lemma** *hcomplex-mult-right-cancel*:  $c \neq 0 \implies a * c = b * c \iff a = b$   
**for**  $a b c :: \text{hcomplex}$   
 ⟨*proof*⟩

## 7.5 Subtraction and Division

**lemma** *hcomplex-diff-eq-eq* [*simp*]:  $x - y = z \iff x = z + y$   
**for**  $x y z :: \text{hcomplex}$   
 ⟨*proof*⟩

## 7.6 Embedding Properties for *hcomplex-of-hypreal* Map

**lemma** *hRe-hcomplex-of-hypreal* [*simp*]:  $\bigwedge z. \text{hRe} (\text{hcomplex-of-hypreal } z) = z$   
 ⟨*proof*⟩

**lemma** *hIm-hcomplex-of-hypreal* [*simp*]:  $\bigwedge z. \text{hIm} (\text{hcomplex-of-hypreal } z) = 0$   
 ⟨*proof*⟩

**lemma** *hcomplex-of-epsilon-not-zero* [*simp*]:  $\text{hcomplex-of-hypreal } \varepsilon \neq 0$   
 ⟨*proof*⟩

## 7.7 *HComplex* theorems

**lemma** *hRe-HComplex* [*simp*]:  $\bigwedge x y. \text{hRe} (\text{HComplex } x y) = x$   
 ⟨*proof*⟩

**lemma** *hIm-HComplex* [*simp*]:  $\bigwedge x y. \text{hIm} (\text{HComplex } x y) = y$   
 ⟨*proof*⟩



**lemma** *hcomplex-surj* [simp]:  $\bigwedge z. HComplex (hRe z) (hIm z) = z$   
 ⟨proof⟩

**lemma** *hcomplex-induct* [case-names rect]:  
 $(\bigwedge x y. P (HComplex x y)) \implies P z$   
 ⟨proof⟩

## 7.8 Modulus (Absolute Value) of Nonstandard Complex Number

**lemma** *hcomplex-of-hypreal-abs*:  
 $hcomplex-of-hypreal |x| = hcomplex-of-hypreal (hcmmod (hcomplex-of-hypreal x))$   
 ⟨proof⟩

**lemma** *HComplex-inject* [simp]:  $\bigwedge x y x' y'. HComplex x y = HComplex x' y' \longleftrightarrow$   
 $x = x' \wedge y = y'$   
 ⟨proof⟩

**lemma** *HComplex-add* [simp]:  
 $\bigwedge x1 y1 x2 y2. HComplex x1 y1 + HComplex x2 y2 = HComplex (x1 + x2) (y1 + y2)$   
 ⟨proof⟩

**lemma** *HComplex-minus* [simp]:  $\bigwedge x y. - HComplex x y = HComplex (- x) (- y)$   
 ⟨proof⟩

**lemma** *HComplex-diff* [simp]:  
 $\bigwedge x1 y1 x2 y2. HComplex x1 y1 - HComplex x2 y2 = HComplex (x1 - x2) (y1 - y2)$   
 ⟨proof⟩

**lemma** *HComplex-mult* [simp]:  
 $\bigwedge x1 y1 x2 y2. HComplex x1 y1 * HComplex x2 y2 = HComplex (x1*x2 - y1*y2)$   
 $(x1*y2 + y1*x2)$   
 ⟨proof⟩

*HComplex-inverse* is proved below.

**lemma** *hcomplex-of-hypreal-eq*:  $\bigwedge r. hcomplex-of-hypreal r = HComplex r 0$   
 ⟨proof⟩

**lemma** *HComplex-add-hcomplex-of-hypreal* [simp]:  
 $\bigwedge x y r. HComplex x y + hcomplex-of-hypreal r = HComplex (x + r) y$   
 ⟨proof⟩

**lemma** *hcomplex-of-hypreal-add-HComplex* [simp]:  
 $\bigwedge r x y. hcomplex-of-hypreal r + HComplex x y = HComplex (r + x) y$   
 ⟨proof⟩

**lemma** *HComplex-mult-hcomplex-of-hypreal*:

$$\bigwedge x y r. HComplex\ x\ y\ * \ hcomplex-of-hypreal\ r = HComplex\ (x\ * \ r)\ (y\ * \ r)$$

*<proof>*

**lemma** *hcomplex-of-hypreal-mult-HComplex*:

$$\bigwedge r\ x\ y. hcomplex-of-hypreal\ r\ * \ HComplex\ x\ y = HComplex\ (r\ * \ x)\ (r\ * \ y)$$

*<proof>*

**lemma** *i-hcomplex-of-hypreal [simp]*:  $\bigwedge r. iii\ * \ hcomplex-of-hypreal\ r = HComplex\ 0\ r$

*<proof>*

**lemma** *hcomplex-of-hypreal-i [simp]*:  $\bigwedge r. hcomplex-of-hypreal\ r\ * \ iii = HComplex\ 0\ r$

*<proof>*

## 7.9 Conjugation

**lemma** *hcomplex-hcnj-cancel-iff [iff]*:  $\bigwedge x\ y. hcnj\ x = hcnj\ y \longleftrightarrow x = y$

*<proof>*

**lemma** *hcomplex-hcnj-hcnj [simp]*:  $\bigwedge z. hcnj\ (hcnj\ z) = z$

*<proof>*

**lemma** *hcomplex-hcnj-hcomplex-of-hypreal [simp]*:

$$\bigwedge x. hcnj\ (hcomplex-of-hypreal\ x) = hcomplex-of-hypreal\ x$$

*<proof>*

**lemma** *hcomplex-hmod-hcnj [simp]*:  $\bigwedge z. hmod\ (hcnj\ z) = hmod\ z$

*<proof>*

**lemma** *hcomplex-hcnj-minus*:  $\bigwedge z. hcnj\ (-\ z) = -\ hcnj\ z$

*<proof>*

**lemma** *hcomplex-hcnj-inverse*:  $\bigwedge z. hcnj\ (inverse\ z) = inverse\ (hcnj\ z)$

*<proof>*

**lemma** *hcomplex-hcnj-add*:  $\bigwedge w\ z. hcnj\ (w + z) = hcnj\ w + hcnj\ z$

*<proof>*

**lemma** *hcomplex-hcnj-diff*:  $\bigwedge w\ z. hcnj\ (w - z) = hcnj\ w - hcnj\ z$

*<proof>*

**lemma** *hcomplex-hcnj-mult*:  $\bigwedge w\ z. hcnj\ (w * z) = hcnj\ w * hcnj\ z$

*<proof>*

**lemma** *hcomplex-hcnj-divide*:  $\bigwedge w\ z. hcnj\ (w / z) = hcnj\ w / hcnj\ z$

*<proof>*

**lemma** *hcnj-one* [simp]:  $hcnj\ 1 = 1$   
 ⟨proof⟩

**lemma** *hcomplex-hcnj-zero* [simp]:  $hcnj\ 0 = 0$   
 ⟨proof⟩

**lemma** *hcomplex-hcnj-zero-iff* [iff]:  $\bigwedge z. hcnj\ z = 0 \longleftrightarrow z = 0$   
 ⟨proof⟩

**lemma** *hcomplex-mult-hcnj*:  $\bigwedge z. z * hcnj\ z = hcomplex-of-hypreal\ ((hRe\ z)^2 + (hIm\ z)^2)$   
 ⟨proof⟩

## 7.10 More Theorems about the Function *hcmmod*

**lemma** *hcmmod-hcomplex-of-hypreal-of-nat* [simp]:  
 $hcmmod\ (hcomplex-of-hypreal\ (hypreal-of-nat\ n)) = hypreal-of-nat\ n$   
 ⟨proof⟩

**lemma** *hcmmod-hcomplex-of-hypreal-of-hypnat* [simp]:  
 $hcmmod\ (hcomplex-of-hypreal\ (hypreal-of-hypnat\ n)) = hypreal-of-hypnat\ n$   
 ⟨proof⟩

**lemma** *hcmmod-mult-hcnj*:  $\bigwedge z. hcmmod\ (z * hcnj\ z) = (hcmmod\ z)^2$   
 ⟨proof⟩

**lemma** *hcmmod-triangle-ineq2* [simp]:  $\bigwedge a\ b. hcmmod\ (b + a) - hcmmod\ b \leq hcmmod\ a$   
 ⟨proof⟩

**lemma** *hcmmod-diff-ineq* [simp]:  $\bigwedge a\ b. hcmmod\ a - hcmmod\ b \leq hcmmod\ (a + b)$   
 ⟨proof⟩

## 7.11 Exponentiation

**lemma** *hcomplexpow-0* [simp]:  $z \hat{=} 0 = 1$   
 for  $z :: hcomplex$   
 ⟨proof⟩

**lemma** *hcomplexpow-Suc* [simp]:  $z \hat{=} (Suc\ n) = z * (z \hat{=} n)$   
 for  $z :: hcomplex$   
 ⟨proof⟩

**lemma** *hcomplexpow-i-squared* [simp]:  $ii^2 = -1$   
 ⟨proof⟩

**lemma** *hcomplex-of-hypreal-pow*:  $\bigwedge x. hcomplex-of-hypreal\ (x \hat{=} n) = hcomplex-of-hypreal\ x \hat{=} n$   
 ⟨proof⟩

**lemma** *hcomplex-hcnj-pow*:  $\bigwedge z. hcnj\ (z \hat{=} n) = hcnj\ z \hat{=} n$

*<proof>*

**lemma** *hcmmod-hcomplexpow*:  $\bigwedge x. \text{hcmmod } (x \wedge n) = \text{hcmmod } x \wedge n$   
*<proof>*

**lemma** *hcpow-minus*:

$\bigwedge x n. (-x :: \text{hcomplex}) \text{ pow } n = (\text{if } (*p* \text{ even}) \text{ then } (x \text{ pow } n) \text{ else } -(x \text{ pow } n))$   
*<proof>*

**lemma** *hcpow-mult*:  $(r * s) \text{ pow } n = (r \text{ pow } n) * (s \text{ pow } n)$   
**for**  $r s :: \text{hcomplex}$   
*<proof>*

**lemma** *hcpow-zero2 [simp]*:  $\bigwedge n. 0 \text{ pow } (\text{hSuc } n) = (0 :: 'a :: \text{semiring-1 star})$   
*<proof>*

**lemma** *hcpow-not-zero [simp,intro]*:  $\bigwedge r n. r \neq 0 \implies r \text{ pow } n \neq (0 :: \text{hcomplex})$   
*<proof>*

**lemma** *hcpow-zero-zero*:  $r \text{ pow } n = 0 \implies r = 0$   
**for**  $r :: \text{hcomplex}$   
*<proof>*

## 7.12 The Function *hsgn*

**lemma** *hsgn-zero [simp]*:  $\text{hsgn } 0 = 0$   
*<proof>*

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

**lemma** *hsgn-minus*:  $\bigwedge z. \text{hsgn } (-z) = -\text{hsgn } z$   
*<proof>*

**lemma** *hsgn-eq*:  $\bigwedge z. \text{hsgn } z = z / \text{hcomplex-of-hypreal } (\text{hcmmod } z)$   
*<proof>*

**lemma** *hcmmod-i*:  $\bigwedge x y. \text{hcmmod } (\text{HComplex } x y) = (*f* \text{ sqrt}) (x^2 + y^2)$   
*<proof>*

**lemma** *hcomplex-eq-cancel-iff1 [simp]*:  
 $\text{hcomplex-of-hypreal } xa = \text{HComplex } x y \longleftrightarrow xa = x \wedge y = 0$   
*<proof>*

**lemma** *hcomplex-eq-cancel-iff2 [simp]*:  
 $\text{HComplex } x y = \text{hcomplex-of-hypreal } xa \longleftrightarrow x = xa \wedge y = 0$   
*<proof>*

**lemma** *HComplex-eq-0* [simp]:  $\bigwedge x y. \text{HComplex } x \ y = 0 \longleftrightarrow x = 0 \wedge y = 0$   
 ⟨proof⟩

**lemma** *HComplex-eq-1* [simp]:  $\bigwedge x y. \text{HComplex } x \ y = 1 \longleftrightarrow x = 1 \wedge y = 0$   
 ⟨proof⟩

**lemma** *i-eq-HComplex-0-1*:  $iii = \text{HComplex } 0 \ 1$   
 ⟨proof⟩

**lemma** *HComplex-eq-i* [simp]:  $\bigwedge x y. \text{HComplex } x \ y = iii \longleftrightarrow x = 0 \wedge y = 1$   
 ⟨proof⟩

**lemma** *hRe-hsgn* [simp]:  $\bigwedge z. \text{hRe } (\text{hsgn } z) = \text{hRe } z / \text{hcm}od \ z$   
 ⟨proof⟩

**lemma** *hIm-hsgn* [simp]:  $\bigwedge z. \text{hIm } (\text{hsgn } z) = \text{hIm } z / \text{hcm}od \ z$   
 ⟨proof⟩

**lemma** *HComplex-inverse*:  $\bigwedge x y. \text{inverse } (\text{HComplex } x \ y) = \text{HComplex } (x / (x^2 + y^2)) \ (-y / (x^2 + y^2))$   
 ⟨proof⟩

**lemma** *hRe-mult-i-eq*[simp]:  $\bigwedge y. \text{hRe } (iii * \text{hcomplex-of-hypreal } y) = 0$   
 ⟨proof⟩

**lemma** *hIm-mult-i-eq* [simp]:  $\bigwedge y. \text{hIm } (iii * \text{hcomplex-of-hypreal } y) = y$   
 ⟨proof⟩

**lemma** *hcm}od-mult-i* [simp]:  $\bigwedge y. \text{hcm}od \ (iii * \text{hcomplex-of-hypreal } y) = |y|$   
 ⟨proof⟩

**lemma** *hcm}od-mult-i2* [simp]:  $\bigwedge y. \text{hcm}od \ (\text{hcomplex-of-hypreal } y * iii) = |y|$   
 ⟨proof⟩

### 7.12.1 *harg*

**lemma** *cos-harg-i-mult-zero* [simp]:  $\bigwedge y. y \neq 0 \implies (*f* \text{cos}) (\text{harg } (\text{HComplex } 0 \ y)) = 0$   
 ⟨proof⟩

## 7.13 Polar Form for Nonstandard Complex Numbers

**lemma** *complex-split-polar2*:  $\forall n. \exists r a. (z \ n) = \text{complex-of-real } r * \text{Complex } (\text{cos } a) \ (\text{sin } a)$   
 ⟨proof⟩

**lemma** *hcomplex-split-polar*:  
 $\bigwedge z. \exists r a. z = \text{hcomplex-of-hypreal } r * (\text{HComplex } ((*f* \text{cos}) \ a) \ ((*f* \text{sin}) \ a))$   
 ⟨proof⟩

**lemma** *hcis-eq*:

$$\bigwedge a. \text{hcis } a = \text{hcomplex-of-hypreal } (( *f* \text{ cos}) a) + \text{iii} * \text{hcomplex-of-hypreal } (( *f* \text{ sin}) a)$$

*<proof>*

**lemma** *hrcis-Ex*:  $\bigwedge z. \exists r a. z = \text{hrcis } r a$

*<proof>*

**lemma** *hRe-hcomplex-polar* [simp]:

$$\bigwedge r a. \text{hRe } (\text{hcomplex-of-hypreal } r * \text{HComplex } (( *f* \text{ cos}) a) (( *f* \text{ sin}) a)) = r * (*f* \text{ cos}) a$$

*<proof>*

**lemma** *hRe-hrcis* [simp]:  $\bigwedge r a. \text{hRe } (\text{hrcis } r a) = r * (*f* \text{ cos}) a$

*<proof>*

**lemma** *hIm-hcomplex-polar* [simp]:

$$\bigwedge r a. \text{hIm } (\text{hcomplex-of-hypreal } r * \text{HComplex } (( *f* \text{ cos}) a) (( *f* \text{ sin}) a)) = r * (*f* \text{ sin}) a$$

*<proof>*

**lemma** *hIm-hrcis* [simp]:  $\bigwedge r a. \text{hIm } (\text{hrcis } r a) = r * (*f* \text{ sin}) a$

*<proof>*

**lemma** *hcmmod-unit-one* [simp]:  $\bigwedge a. \text{hcmmod } (\text{HComplex } (( *f* \text{ cos}) a) (( *f* \text{ sin}) a)) = 1$

*<proof>*

**lemma** *hcmmod-complex-polar* [simp]:

$$\bigwedge r a. \text{hcmmod } (\text{hcomplex-of-hypreal } r * \text{HComplex } (( *f* \text{ cos}) a) (( *f* \text{ sin}) a)) = |r|$$

*<proof>*

**lemma** *hcmmod-hrcis* [simp]:  $\bigwedge r a. \text{hcmmod}(\text{hrcis } r a) = |r|$

*<proof>*

$$(r1 * \text{hrcis } a) * (r2 * \text{hrcis } b) = r1 * r2 * \text{hrcis } (a + b)$$

**lemma** *hcis-hrcis-eq*:  $\bigwedge a. \text{hcis } a = \text{hrcis } 1 a$

*<proof>*

**declare** *hcis-hrcis-eq* [symmetric, simp]

**lemma** *hrcis-mult*:  $\bigwedge a b r1 r2. \text{hrcis } r1 a * \text{hrcis } r2 b = \text{hrcis } (r1 * r2) (a + b)$

*<proof>*

**lemma** *hcis-mult*:  $\bigwedge a b. \text{hcis } a * \text{hcis } b = \text{hcis } (a + b)$

*<proof>*

**lemma** *hcis-zero* [simp]:  $\text{hcis } 0 = 1$

*<proof>*

**lemma** *hrcis-zero-mod* [simp]:  $\bigwedge a. \text{hrcis } 0 \ a = 0$   
 ⟨proof⟩

**lemma** *hrcis-zero-arg* [simp]:  $\bigwedge r. \text{hrcis } r \ 0 = \text{hcomplex-of-hypreal } r$   
 ⟨proof⟩

**lemma** *hcomplex-i-mult-minus* [simp]:  $\bigwedge x. \text{iii} * (\text{iii} * x) = - x$   
 ⟨proof⟩

**lemma** *hcomplex-i-mult-minus2* [simp]:  $\text{iii} * \text{iii} * x = - x$   
 ⟨proof⟩

**lemma** *hcis-hypreal-of-nat-Suc-mult*:  
 $\bigwedge a. \text{hcis } (\text{hypreal-of-nat } (\text{Suc } n) * a) = \text{hcis } a * \text{hcis } (\text{hypreal-of-nat } n * a)$   
 ⟨proof⟩

**lemma** *NSDeMoivre*:  $\bigwedge a. (\text{hcis } a) ^ n = \text{hcis } (\text{hypreal-of-nat } n * a)$   
 ⟨proof⟩

**lemma** *hcis-hypreal-of-hypnat-Suc-mult*:  
 $\bigwedge a \ n. \text{hcis } (\text{hypreal-of-hypnat } (n + 1) * a) = \text{hcis } a * \text{hcis } (\text{hypreal-of-hypnat } n * a)$   
 ⟨proof⟩

**lemma** *NSDeMoivre-ext*:  $\bigwedge a \ n. (\text{hcis } a) \ \text{pow } n = \text{hcis } (\text{hypreal-of-hypnat } n * a)$   
 ⟨proof⟩

**lemma** *NSDeMoivre2*:  $\bigwedge a \ r. (\text{hrcis } r \ a) ^ n = \text{hrcis } (r ^ n) (\text{hypreal-of-nat } n * a)$   
 ⟨proof⟩

**lemma** *DeMoivre2-ext*:  $\bigwedge a \ r \ n. (\text{hrcis } r \ a) \ \text{pow } n = \text{hrcis } (r \ \text{pow } n) (\text{hypreal-of-hypnat } n * a)$   
 ⟨proof⟩

**lemma** *hcis-inverse* [simp]:  $\bigwedge a. \text{inverse } (\text{hcis } a) = \text{hcis } (- a)$   
 ⟨proof⟩

**lemma** *hrcis-inverse*:  $\bigwedge a \ r. \text{inverse } (\text{hrcis } r \ a) = \text{hrcis } (\text{inverse } r) (- a)$   
 ⟨proof⟩

**lemma** *hRe-hcis* [simp]:  $\bigwedge a. \text{hRe } (\text{hcis } a) = (*f* \ \text{cos}) \ a$   
 ⟨proof⟩

**lemma** *hIm-hcis* [simp]:  $\bigwedge a. \text{hIm } (\text{hcis } a) = (*f* \ \text{sin}) \ a$   
 ⟨proof⟩

**lemma** *cos-n-hRe-hcis-pow-n*:  $(*f* \ \text{cos}) (\text{hypreal-of-nat } n * a) = \text{hRe } (\text{hcis } a ^ n)$   
 ⟨proof⟩

**lemma** *sin-n-hIm-hcis-pow-n*: ( $*f*$  *sin*) (*hypreal-of-nat*  $n * a$ ) = *hIm* (*hcis*  $a \hat{\ } n$ )  
 ⟨*proof*⟩

**lemma** *cos-n-hRe-hcis-hcpow-n*: ( $*f*$  *cos*) (*hypreal-of-hypnat*  $n * a$ ) = *hRe* (*hcis*  $a \text{ pow } n$ )  
 ⟨*proof*⟩

**lemma** *sin-n-hIm-hcis-hcpow-n*: ( $*f*$  *sin*) (*hypreal-of-hypnat*  $n * a$ ) = *hIm* (*hcis*  $a \text{ pow } n$ )  
 ⟨*proof*⟩

**lemma** *hExp-add*:  $\bigwedge a b. \text{hExp } (a + b) = \text{hExp } a * \text{hExp } b$   
 ⟨*proof*⟩

### 7.14 *hcomplex-of-complex*: the Injection from type *complex* to *hcomplex*

**lemma** *hcomplex-of-complex-i*: *iii* = *hcomplex-of-complex i*  
 ⟨*proof*⟩

**lemma** *hRe-hcomplex-of-complex*: *hRe* (*hcomplex-of-complex*  $z$ ) = *hypreal-of-real* (*Re*  $z$ )  
 ⟨*proof*⟩

**lemma** *hIm-hcomplex-of-complex*: *hIm* (*hcomplex-of-complex*  $z$ ) = *hypreal-of-real* (*Im*  $z$ )  
 ⟨*proof*⟩

**lemma** *hcmmod-hcomplex-of-complex*: *hcmmod* (*hcomplex-of-complex*  $x$ ) = *hypreal-of-real* (*cmmod*  $x$ )  
 ⟨*proof*⟩

### 7.15 Numerals and Arithmetic

**lemma** *hcomplex-of-hypreal-eq-hcomplex-of-complex*:  
*hcomplex-of-hypreal* (*hypreal-of-real*  $x$ ) = *hcomplex-of-complex* (*complex-of-real*  $x$ )  
 ⟨*proof*⟩

**lemma** *hcomplex-hypreal-numeral*:  
*hcomplex-of-complex* (*numeral*  $w$ ) = *hcomplex-of-hypreal*(*numeral*  $w$ )  
 ⟨*proof*⟩

**lemma** *hcomplex-hypreal-neg-numeral*:  
*hcomplex-of-complex* ( $- \text{numeral } w$ ) = *hcomplex-of-hypreal*( $- \text{numeral } w$ )  
 ⟨*proof*⟩

**lemma** *hcomplex-numeral-hcnj* [*simp*]: *hcnj* (*numeral*  $v :: \text{hcomplex}$ ) = *numeral*  $v$   
 ⟨*proof*⟩



**lemma** *hcomplex-numeral-hcmod* [simp]:  $hcmod (numeral\ v :: hcomplex) = (numeral\ v :: hypreal)$   
 ⟨proof⟩

**lemma** *hcomplex-neg-numeral-hcmod* [simp]:  $hcmod (-\ numeral\ v :: hcomplex) = (numeral\ v :: hypreal)$   
 ⟨proof⟩

**lemma** *hcomplex-numeral-hRe* [simp]:  $hRe (numeral\ v :: hcomplex) = numeral\ v$   
 ⟨proof⟩

**lemma** *hcomplex-numeral-hIm* [simp]:  $hIm (numeral\ v :: hcomplex) = 0$   
 ⟨proof⟩

end

## 8 Star-Transforms in Non-Standard Analysis

**theory** *Star*  
 imports *NSA*  
 begin

**definition** — internal sets  
*starset-n* ::  $(nat \Rightarrow 'a\ set) \Rightarrow 'a\ star\ set$  (*\*sn\** - [80] 80)  
 where *\*sn\**  $As = Iset (star-n\ As)$

**definition** *InternalSets* ::  $'a\ star\ set\ set$   
 where *InternalSets* =  $\{X. \exists As. X = *sn*\ As\}$

**definition** — nonstandard extension of function  
*is-starext* ::  $('a\ star \Rightarrow 'a\ star) \Rightarrow ('a \Rightarrow 'a) \Rightarrow bool$   
 where *is-starext*  $F\ f \longleftrightarrow$   
 $(\forall x\ y. \exists X \in Rep-star\ x. \exists Y \in Rep-star\ y. y = F\ x \longleftrightarrow eventually\ (\lambda n. Y\ n = f(X\ n))\ \mathcal{U})$

**definition** — internal functions  
*starfun-n* ::  $(nat \Rightarrow 'a \Rightarrow 'b) \Rightarrow 'a\ star \Rightarrow 'b\ star$  (*\*fn\** - [80] 80)  
 where *\*fn\**  $F = Ifun (star-n\ F)$

**definition** *InternalFuns* ::  $('a\ star \Rightarrow 'b\ star)\ set$   
 where *InternalFuns* =  $\{X. \exists F. X = *fn*\ F\}$

### 8.1 Preamble - Pulling $\exists$ over $\forall$

This proof does not need AC and was suggested by the referee for the JCM Paper: let  $f\ x$  be least  $y$  such that  $Q\ x\ y$ .

**lemma** *no-choice*:  $\forall x. \exists y. Q\ x\ y \implies \exists f :: 'a \Rightarrow nat. \forall x. Q\ x\ (f\ x)$

*<proof>*

## 8.2 Properties of the Star-transform Applied to Sets of Reals

**lemma** *STAR-star-of-image-subset*:  $\text{star-of } ' A \subseteq *s* A$

*<proof>*

**lemma** *STAR-hypreal-of-real-Int*:  $*s* X \cap \mathbb{R} = \text{hypreal-of-real } ' X$

*<proof>*

**lemma** *STAR-star-of-Int*:  $*s* X \cap \text{Standard} = \text{star-of } ' X$

*<proof>*

**lemma** *lemma-not-hyprealA*:  $x \notin \text{hypreal-of-real } ' A \implies \forall y \in A. x \neq \text{hypreal-of-real } y$

*<proof>*

**lemma** *lemma-not-starA*:  $x \notin \text{star-of } ' A \implies \forall y \in A. x \neq \text{star-of } y$

*<proof>*

**lemma** *STAR-real-seq-to-hypreal*:  $\forall n. (X n) \notin M \implies \text{star-n } X \notin *s* M$

*<proof>*

**lemma** *STAR-singleton*:  $*s* \{x\} = \{\text{star-of } x\}$

*<proof>*

**lemma** *STAR-not-mem*:  $x \notin F \implies \text{star-of } x \notin *s* F$

*<proof>*

**lemma** *STAR-subset-closed*:  $x \in *s* A \implies A \subseteq B \implies x \in *s* B$

*<proof>*

Nonstandard extension of a set (defined using a constant sequence) as a special case of an internal set.

**lemma** *starset-n-starset*:  $\forall n. A s n = A \implies *sn* A s = *s* A$

*<proof>*

## 8.3 Theorems about nonstandard extensions of functions

Nonstandard extension of a function (defined using a constant sequence) as a special case of an internal function.

**lemma** *starfun-n-starfun*:  $F = (\lambda n. f) \implies *fn* F = *f* f$

*<proof>*

Prove that *abs* for hypreal is a nonstandard extension of *abs* for real w/o use of congruence property (proved after this for general nonstandard extensions of real valued functions).

Proof now Uses the ultrafilter tactic!

**lemma** *hrabs-is-starext-rabs: is-starext abs abs*  
 ⟨*proof*⟩

Nonstandard extension of functions.

**lemma** *starfun: ( \*f\* f) (star-n X) = star-n (λn. f (X n))*  
 ⟨*proof*⟩

**lemma** *starfun-if-eq: Λw. w ≠ star-of x ⇒ ( \*f\* (λz. if z = x then a else g z))*  
*w = ( \*f\* g) w*  
 ⟨*proof*⟩

Multiplication: ( \*f) x ( \*g) = \*(f x g)

**lemma** *starfun-mult: Λx. ( \*f\* f) x \* ( \*f\* g) x = ( \*f\* (λx. f x \* g x)) x*  
 ⟨*proof*⟩

**declare** *starfun-mult* [*symmetric, simp*]

Addition: ( \*f) + ( \*g) = \*(f + g)

**lemma** *starfun-add: Λx. ( \*f\* f) x + ( \*f\* g) x = ( \*f\* (λx. f x + g x)) x*  
 ⟨*proof*⟩

**declare** *starfun-add* [*symmetric, simp*]

Subtraction: ( \*f) + -( \*g) = \*(f + -g)

**lemma** *starfun-minus: Λx. - ( \*f\* f) x = ( \*f\* (λx. - f x)) x*  
 ⟨*proof*⟩

**declare** *starfun-minus* [*symmetric, simp*]

**lemma** *starfun-add-minus: Λx. ( \*f\* f) x + -( \*f\* g) x = ( \*f\* (λx. f x + -g x)) x*  
 ⟨*proof*⟩

**declare** *starfun-add-minus* [*symmetric, simp*]

**lemma** *starfun-diff: Λx. ( \*f\* f) x - ( \*f\* g) x = ( \*f\* (λx. f x - g x)) x*  
 ⟨*proof*⟩

**declare** *starfun-diff* [*symmetric, simp*]

Composition: ( \*f) ○ ( \*g) = \*(f ○ g)

**lemma** *starfun-o2: (λx. ( \*f\* f) (( \*f\* g) x)) = \*f\* (λx. f (g x))*  
 ⟨*proof*⟩

**lemma** *starfun-o: ( \*f\* f) ○ ( \*f\* g) = ( \*f\* (f ○ g))*  
 ⟨*proof*⟩

NS extension of constant function.

**lemma** *starfun-const-fun* [*simp*]: Λx. ( \*f\* (λx. k)) x = *star-of k*  
 ⟨*proof*⟩

The NS extension of the identity function.

**lemma** *starfun-Id* [*simp*]:  $\bigwedge x. (*f* (\lambda x. x)) x = x$   
 ⟨*proof*⟩

The Star-function is a (nonstandard) extension of the function.

**lemma** *is-starext-starfun*: *is-starext* (*\*f\* f*) *f*  
 ⟨*proof*⟩

Any nonstandard extension is in fact the Star-function.

**lemma** *is-starfun-starext*:  
**assumes** *is-starext F f*  
**shows**  $F = *f* f$   
 ⟨*proof*⟩

**lemma** *is-starext-starfun-iff*: *is-starext F f*  $\longleftrightarrow F = *f* f$   
 ⟨*proof*⟩

Extended function has same solution as its standard version for real arguments. i.e they are the same for all real arguments.

**lemma** *starfun-eq*: (*\*f\* f*) (*star-of a*) = *star-of (f a)*  
 ⟨*proof*⟩

**lemma** *starfun-approx*: (*\*f\* f*) (*star-of a*)  $\approx$  *star-of (f a)*  
 ⟨*proof*⟩

Useful for NS definition of derivatives.

**lemma** *starfun-lambda-cancel*:  $\bigwedge x'. (*f* (\lambda h. f (x + h))) x' = (*f* f) (star-of x + x')$   
 ⟨*proof*⟩

**lemma** *starfun-lambda-cancel2*: (*\*f\* (\lambda h. f (g (x + h)))*)  $x' = (*f* (f \circ g)) (star-of x + x')$   
 ⟨*proof*⟩

**lemma** *starfun-mult-HFinite-approx*:  
 (*\*f\* f*)  $x \approx l \implies (*f* g) x \approx m \implies l \in HFinite \implies m \in HFinite \implies$   
 (*\*f\* (\lambda x. f x \* g x)*)  $x \approx l * m$   
**for**  $l m :: 'a::real-normed-algebra$  *star*  
 ⟨*proof*⟩

**lemma** *starfun-add-approx*: (*\*f\* f*)  $x \approx l \implies (*f* g) x \approx m \implies (*f* (\%x. f x + g x)) x \approx l + m$   
 ⟨*proof*⟩

Examples: *hrabs* is nonstandard extension of *rabs*, *inverse* is nonstandard extension of *inverse*.

Can be proved easily using theorem *starfun* and properties of ultrafilter as for *inverse* below we use the theorem we proved above instead.

**lemma** *starfun-rabs-hrabs*:  $*f* \text{ abs} = \text{abs}$   
 ⟨*proof*⟩

**lemma** *starfun-inverse-inverse* [*simp*]:  $( *f* \text{ inverse} ) x = \text{inverse } x$   
 ⟨*proof*⟩

**lemma** *starfun-inverse*:  $\bigwedge x. \text{inverse } (( *f* f ) x) = ( *f* (\lambda x. \text{inverse } (f x))) x$   
 ⟨*proof*⟩

**declare** *starfun-inverse* [*symmetric, simp*]

**lemma** *starfun-divide*:  $\bigwedge x. ( *f* f ) x / ( *f* g ) x = ( *f* (\lambda x. f x / g x) ) x$   
 ⟨*proof*⟩

**declare** *starfun-divide* [*symmetric, simp*]

**lemma** *starfun-inverse2*:  $\bigwedge x. \text{inverse } (( *f* f ) x) = ( *f* (\lambda x. \text{inverse } (f x))) x$   
 ⟨*proof*⟩

General lemma/theorem needed for proofs in elementary topology of the reals.

**lemma** *starfun-mem-starset*:  $\bigwedge x. ( *f* f ) x \in *s* A \implies x \in *s* \{x. f x \in A\}$   
 ⟨*proof*⟩

Alternative definition for *hrabs* with *rabs* function applied entrywise to equivalence class representative. This is easily proved using *starfun* and *ns* extension thm.

**lemma** *hypreal-hrabs*:  $|star-n X| = star-n (\lambda n. |X n|)$   
 ⟨*proof*⟩

Nonstandard extension of set through nonstandard extension of *rabs* function i.e. *hrabs*. A more general result should be where we replace *rabs* by some arbitrary function *f* and *hrabs* by its NS extension. See second NS set extension below.

**lemma** *STAR-rabs-add-minus*:  $*s* \{x. |x + - y| < r\} = \{x. |x + -star-of y| < star-of r\}$   
 ⟨*proof*⟩

**lemma** *STAR-starfun-rabs-add-minus*:  
 $*s* \{x. |f x + - y| < r\} = \{x. |( *f* f ) x + -star-of y| < star-of r\}$   
 ⟨*proof*⟩

Another characterization of Infinitesimal and one of  $\approx$  relation. In this theory since *hypreal-hrabs* proved here. Maybe move both theorems??

**lemma** *Infinitesimal-FreeUltrafilterNat-iff2*:  
 $star-n X \in \text{Infinitesimal} \iff (\forall m. \text{eventually } (\lambda n. \text{norm } (X n) < \text{inverse } (\text{real } (Suc m)))) \mathcal{U}$   
 ⟨*proof*⟩

**lemma** *HNatInfinite-inverse-Infinitesimal* [*simp*]:  
**assumes**  $n \in \text{HNatInfinite}$   
**shows**  $\text{inverse}(\text{hypreal-of-hypnat } n) \in \text{Infinitesimal}$   
 $\langle \text{proof} \rangle$

**lemma** *approx-FreeUltrafilterNat-iff*:  
 $\text{star-}n \ X \approx \text{star-}n \ Y \iff (\forall r > 0. \text{eventually } (\lambda n. \text{norm}(X \ n - Y \ n) < r) \ \mathcal{U})$   
**(is ?lhs = ?rhs)**  
 $\langle \text{proof} \rangle$

**lemma** *approx-FreeUltrafilterNat-iff2*:  
 $\text{star-}n \ X \approx \text{star-}n \ Y \iff (\forall m. \text{eventually } (\lambda n. \text{norm}(X \ n - Y \ n) < \text{inverse}(\text{real}(\text{Suc } m))) \ \mathcal{U})$   
**(is ?lhs = ?rhs)**  
 $\langle \text{proof} \rangle$

**lemma** *inj-starfun*:  $\text{inj starfun}$   
 $\langle \text{proof} \rangle$

**end**

## 9 Star-transforms for the Hypernaturals

**theory** *NatStar*  
**imports** *Star*  
**begin**

**lemma** *star-n-eq-starfun-whn*:  $\text{star-}n \ X = (*f* \ X) \ \text{whn}$   
 $\langle \text{proof} \rangle$

**lemma** *starset-n-Un*:  $*sn* (\lambda n. (A \ n) \cup (B \ n)) = *sn* \ A \cup *sn* \ B$   
 $\langle \text{proof} \rangle$

**lemma** *InternalSets-Un*:  $X \in \text{InternalSets} \implies Y \in \text{InternalSets} \implies X \cup Y \in \text{InternalSets}$   
 $\langle \text{proof} \rangle$

**lemma** *starset-n-Int*:  $*sn* (\lambda n. A \ n \cap B \ n) = *sn* \ A \cap *sn* \ B$   
 $\langle \text{proof} \rangle$

**lemma** *InternalSets-Int*:  $X \in \text{InternalSets} \implies Y \in \text{InternalSets} \implies X \cap Y \in \text{InternalSets}$   
 $\langle \text{proof} \rangle$

**lemma** *starset-n-Compl*:  $*sn* ((\lambda n. - A \ n)) = - (*sn* \ A)$   
 $\langle \text{proof} \rangle$

**lemma** *InternalSets-Compl*:  $X \in \text{InternalSets} \implies - X \in \text{InternalSets}$   
 $\langle \text{proof} \rangle$

**lemma** *starset-n-diff*:  $*sn* (\lambda n. (A\ n) - (B\ n)) = *sn* A - *sn* B$   
 ⟨proof⟩

**lemma** *InternalSets-diff*:  $X \in InternalSets \implies Y \in InternalSets \implies X - Y \in InternalSets$   
 ⟨proof⟩

**lemma** *NatStar-SHNat-subset*:  $Nats \leq *s* (UNIV:: nat\ set)$   
 ⟨proof⟩

**lemma** *NatStar-hypreal-of-real-Int*:  $*s* X\ Int\ Nats = hypnat-of-nat\ ` X$   
 ⟨proof⟩

**lemma** *starset-starset-n-eq*:  $*s* X = *sn* (\lambda n. X)$   
 ⟨proof⟩

**lemma** *InternalSets-starset-n [simp]*:  $( *s* X ) \in InternalSets$   
 ⟨proof⟩

**lemma** *InternalSets-UNIV-diff*:  $X \in InternalSets \implies UNIV - X \in InternalSets$   
 ⟨proof⟩

## 9.1 Nonstandard Extensions of Functions

Example of transfer of a property from reals to hyperreals — used for limit comparison of sequences.

**lemma** *starfun-le-mono*:  $\forall n. N \leq n \longrightarrow f\ n \leq g\ n \implies \forall n. hypnat-of-nat\ N \leq n \longrightarrow ( *f* f )\ n \leq ( *f* g )\ n$   
 ⟨proof⟩

And another:

**lemma** *starfun-less-mono*:  
 $\forall n. N \leq n \longrightarrow f\ n < g\ n \implies \forall n. hypnat-of-nat\ N \leq n \longrightarrow ( *f* f )\ n < ( *f* g )\ n$   
 ⟨proof⟩

Nonstandard extension when we increment the argument by one.

**lemma** *starfun-shift-one*:  $\bigwedge N. ( *f* (\lambda n. f\ (Suc\ n)) )\ N = ( *f* f )\ (N + (1::hypnat))$   
 ⟨proof⟩

Nonstandard extension with absolute value.

**lemma** *starfun-abs*:  $\bigwedge N. ( *f* (\lambda n. |f\ n|) )\ N = |( *f* f )\ N|$   
 ⟨proof⟩

The *hyperpow* function as a nonstandard extension of *realpow*.

**lemma** *starfun-pow*:  $\bigwedge N. ( *f* (\lambda n. r \hat{\ } n) )\ N = hypreal-of-real\ r\ pow\ N$   
 ⟨proof⟩

**lemma** *starfun-pow2*:  $\bigwedge N. (*f* (\lambda n. X n \hat{=} m)) N = (*f* X) N \text{ pow hypnat-of-nat } m$   
 ⟨proof⟩

**lemma** *starfun-pow3*:  $\bigwedge R. (*f* (\lambda r. r \hat{=} n)) R = R \text{ pow hypnat-of-nat } n$   
 ⟨proof⟩

The *hypreal-of-hypnat* function as a nonstandard extension of *real*.

**lemma** *starfunNat-real-of-nat*:  $(*f* \text{ real}) = \text{hypreal-of-hypnat}$   
 ⟨proof⟩

**lemma** *starfun-inverse-real-of-nat-eq*:  
 $N \in \text{HNatInfinite} \implies (*f* (\lambda x::\text{nat}. \text{inverse} (\text{real } x))) N = \text{inverse} (\text{hypreal-of-hypnat } N)$   
 ⟨proof⟩

Internal functions – some redundancy with *\*f\** now.

**lemma** *starfun-n*:  $(*fn* f) (\text{star-n } X) = \text{star-n } (\lambda n. f n (X n))$   
 ⟨proof⟩

Multiplication:  $(*fn) x (*gn) = *(fn x gn)$

**lemma** *starfun-n-mult*:  $(*fn* f) z * (*fn* g) z = (*fn* (\lambda i x. f i x * g i x)) z$   
 ⟨proof⟩

Addition:  $(*fn) + (*gn) = *(fn + gn)$

**lemma** *starfun-n-add*:  $(*fn* f) z + (*fn* g) z = (*fn* (\lambda i x. f i x + g i x)) z$   
 ⟨proof⟩

Subtraction:  $(*fn) - (*gn) = *(fn + - gn)$

**lemma** *starfun-n-add-minus*:  $(*fn* f) z + -( *fn* g) z = (*fn* (\lambda i x. f i x + -g i x)) z$   
 ⟨proof⟩

Composition:  $(*fn) \circ (*gn) = *(fn \circ gn)$

**lemma** *starfun-n-const-fun [simp]*:  $(*fn* (\lambda i x. k)) z = \text{star-of } k$   
 ⟨proof⟩

**lemma** *starfun-n-minus*:  $-( *fn* f) x = (*fn* (\lambda i x. -(f i) x)) x$   
 ⟨proof⟩

**lemma** *starfun-n-eq [simp]*:  $(*fn* f) (\text{star-of } n) = \text{star-n } (\lambda i. f i n)$   
 ⟨proof⟩

**lemma** *starfun-eq-iff*:  $(( *f* f) = (*f* g)) \longleftrightarrow f = g$   
 ⟨proof⟩

**lemma** *starfunNat-inverse-real-of-nat-Infinitesimal [simp]*:



$N \in \mathit{HNatInfinite} \implies (*f* (\lambda x. \mathit{inverse} (\mathit{real} x))) N \in \mathit{Infinitesimal}$   
 ⟨proof⟩

## 9.2 Nonstandard Characterization of Induction

**lemma** *hypnat-induct-obj*:

$\bigwedge n. ((*p* P) (0::\mathit{hypnat}) \wedge (\forall n. (*p* P) n \longrightarrow (*p* P) (n + 1))) \longrightarrow (*p* P) n$   
 ⟨proof⟩

**lemma** *hypnat-induct*:

$\bigwedge n. (*p* P) (0::\mathit{hypnat}) \implies (\bigwedge n. (*p* P) n \implies (*p* P) (n + 1)) \implies (*p* P) n$   
 ⟨proof⟩

**lemma** *starP2-eq-iff*:  $(*p2* (=)) = (=)$

⟨proof⟩

**lemma** *starP2-eq-iff2*:  $(*p2* (\lambda x y. x = y)) X Y \longleftrightarrow X = Y$

⟨proof⟩

**lemma** *nonempty-set-star-has-least-lemma*:

$\exists n \in S. \forall m \in S. n \leq m$  **if**  $S \neq \{\}$  **for**  $S :: \mathit{nat} \mathit{set}$

⟨proof⟩

**lemma** *nonempty-set-star-has-least*:

$\bigwedge S::\mathit{nat} \mathit{set} \mathit{star}. \mathit{Iset} S \neq \{\} \implies \exists n \in \mathit{Iset} S. \forall m \in \mathit{Iset} S. n \leq m$

⟨proof⟩

**lemma** *nonempty-InternalNatSet-has-least*:  $S \in \mathit{InternalSets} \implies S \neq \{\} \implies \exists n \in S. \forall m \in S. n \leq m$

**for**  $S :: \mathit{hypnat} \mathit{set}$

⟨proof⟩

Goldblatt, page 129 Thm 11.3.2.

**lemma** *internal-induct-lemma*:

$\bigwedge X::\mathit{nat} \mathit{set} \mathit{star}.$

$(0::\mathit{hypnat}) \in \mathit{Iset} X \implies \forall n. n \in \mathit{Iset} X \longrightarrow n + 1 \in \mathit{Iset} X \implies \mathit{Iset} X =$

$(\mathit{UNIV}::\mathit{hypnat} \mathit{set})$

⟨proof⟩

**lemma** *internal-induct*:

$X \in \mathit{InternalSets} \implies (0::\mathit{hypnat}) \in X \implies \forall n. n \in X \longrightarrow n + 1 \in X \implies X =$

$(\mathit{UNIV}::\mathit{hypnat} \mathit{set})$

⟨proof⟩

**end**

## 10 Sequences and Convergence (Nonstandard)

**theory** *HSEQ*

**imports** *Complex-Main NatStar*

**abbrevs**  $----> = \longrightarrow_{NS}$

**begin**

**definition** *NSLIMSEQ* ::  $(nat \Rightarrow 'a::real-normed-vector) \Rightarrow 'a \Rightarrow bool$

$(((-)/ \longrightarrow_{NS} (-)) [60, 60] 60)$  **where**

— Nonstandard definition of convergence of sequence

$X \longrightarrow_{NS} L \iff (\forall N \in HNatInfinite. (*f* X) N \approx star-of L)$

**definition** *nslim* ::  $(nat \Rightarrow 'a::real-normed-vector) \Rightarrow 'a$

**where** *nslim*  $X = (THE L. X \longrightarrow_{NS} L)$

— Nonstandard definition of limit using choice operator

**definition** *NSconvergent* ::  $(nat \Rightarrow 'a::real-normed-vector) \Rightarrow bool$

**where** *NSconvergent*  $X \iff (\exists L. X \longrightarrow_{NS} L)$

— Nonstandard definition of convergence

**definition** *NSBseq* ::  $(nat \Rightarrow 'a::real-normed-vector) \Rightarrow bool$

**where** *NSBseq*  $X \iff (\forall N \in HNatInfinite. (*f* X) N \in HFinite)$

— Nonstandard definition for bounded sequence

**definition** *NSCauchy* ::  $(nat \Rightarrow 'a::real-normed-vector) \Rightarrow bool$

**where** *NSCauchy*  $X \iff (\forall M \in HNatInfinite. \forall N \in HNatInfinite. (*f* X) M \approx (*f* X) N)$

— Nonstandard definition

### 10.1 Limits of Sequences

**lemma** *NSLIMSEQ-I*:  $(\bigwedge N. N \in HNatInfinite \implies starfun X N \approx star-of L) \implies X \longrightarrow_{NS} L$

*<proof>*

**lemma** *NSLIMSEQ-D*:  $X \longrightarrow_{NS} L \implies N \in HNatInfinite \implies starfun X N \approx star-of L$

*<proof>*

**lemma** *NSLIMSEQ-const*:  $(\lambda n. k) \longrightarrow_{NS} k$

*<proof>*

**lemma** *NSLIMSEQ-add*:  $X \longrightarrow_{NS} a \implies Y \longrightarrow_{NS} b \implies (\lambda n. X n + Y n) \longrightarrow_{NS} a + b$

*<proof>*

**lemma** *NSLIMSEQ-add-const*:  $f \longrightarrow_{NS} a \implies (\lambda n. f n + b) \longrightarrow_{NS} a + b$

*<proof>*

**lemma** *NSLIMSEQ-mult*:  $X \longrightarrow_{NS} a \implies Y \longrightarrow_{NS} b \implies (\lambda n. X\ n * Y\ n) \longrightarrow_{NS} a * b$   
**for**  $a\ b :: 'a::real-normed-algebra$   
*<proof>*

**lemma** *NSLIMSEQ-minus*:  $X \longrightarrow_{NS} a \implies (\lambda n. - X\ n) \longrightarrow_{NS} - a$   
*<proof>*

**lemma** *NSLIMSEQ-minus-cancel*:  $(\lambda n. - X\ n) \longrightarrow_{NS} - a \implies X \longrightarrow_{NS} a$   
*<proof>*

**lemma** *NSLIMSEQ-diff*:  $X \longrightarrow_{NS} a \implies Y \longrightarrow_{NS} b \implies (\lambda n. X\ n - Y\ n) \longrightarrow_{NS} a - b$   
*<proof>*

**lemma** *NSLIMSEQ-diff-const*:  $f \longrightarrow_{NS} a \implies (\lambda n. f\ n - b) \longrightarrow_{NS} a - b$   
*<proof>*

**lemma** *NSLIMSEQ-inverse*:  $X \longrightarrow_{NS} a \implies a \neq 0 \implies (\lambda n. inverse\ (X\ n)) \longrightarrow_{NS} inverse\ a$   
**for**  $a :: 'a::real-normed-div-algebra$   
*<proof>*

**lemma** *NSLIMSEQ-mult-inverse*:  $X \longrightarrow_{NS} a \implies Y \longrightarrow_{NS} b \implies b \neq 0 \implies (\lambda n. X\ n / Y\ n) \longrightarrow_{NS} a / b$   
**for**  $a\ b :: 'a::real-normed-field$   
*<proof>*

**lemma** *starfun-hnorm*:  $\bigwedge x. hnorm\ (( *f* f)\ x) = ( *f* (\lambda x. norm\ (f\ x)))\ x$   
*<proof>*

**lemma** *NSLIMSEQ-norm*:  $X \longrightarrow_{NS} a \implies (\lambda n. norm\ (X\ n)) \longrightarrow_{NS} norm\ a$   
*<proof>*

Uniqueness of limit.

**lemma** *NSLIMSEQ-unique*:  $X \longrightarrow_{NS} a \implies X \longrightarrow_{NS} b \implies a = b$   
*<proof>*

**lemma** *NSLIMSEQ-pow* [rule-format]:  $(X \longrightarrow_{NS} a) \longrightarrow ((\lambda n. (X\ n) \wedge m) \longrightarrow_{NS} a \wedge m)$   
**for**  $a :: 'a::\{real-normed-algebra,power\}$   
*<proof>*

We can now try and derive a few properties of sequences, starting with the limit comparison property for sequences.

**lemma** *NSLIMSEQ-le*:  $f \longrightarrow_{NS} l \implies g \longrightarrow_{NS} m \implies \exists N. \forall n \geq N. f\ n \leq g\ n \implies l \leq m$

**for**  $l\ m :: \text{real}$   
 ⟨*proof*⟩

**lemma** *NSLIMSEQ-le-const*:  $X \longrightarrow_{NS} r \implies \forall n. a \leq X\ n \implies a \leq r$   
**for**  $a\ r :: \text{real}$   
 ⟨*proof*⟩

**lemma** *NSLIMSEQ-le-const2*:  $X \longrightarrow_{NS} r \implies \forall n. X\ n \leq a \implies r \leq a$   
**for**  $a\ r :: \text{real}$   
 ⟨*proof*⟩

Shift a convergent series by 1: By the equivalence between Cauchiness and convergence and because the successor of an infinite hypernatural is also infinite.

**lemma** *NSLIMSEQ-Suc-iff*:  $((\lambda n. f\ (Suc\ n)) \longrightarrow_{NS} l) \longleftrightarrow (f \longrightarrow_{NS} l)$   
 ⟨*proof*⟩

### 10.1.1 Equivalence of LIMSEQ and NSLIMSEQ

**lemma** *LIMSEQ-NSLIMSEQ*:  
**assumes**  $X: X \longrightarrow L$   
**shows**  $X \longrightarrow_{NS} L$   
 ⟨*proof*⟩

**lemma** *NSLIMSEQ-LIMSEQ*:  
**assumes**  $X: X \longrightarrow_{NS} L$   
**shows**  $X \longrightarrow L$   
 ⟨*proof*⟩

**theorem** *LIMSEQ-NSLIMSEQ-iff*:  $f \longrightarrow L \longleftrightarrow f \longrightarrow_{NS} L$   
 ⟨*proof*⟩

### 10.1.2 Derived theorems about NSLIMSEQ

We prove the NS version from the standard one, since the NS proof seems more complicated than the standard one above!

**lemma** *NSLIMSEQ-norm-zero*:  $(\lambda n. \text{norm}\ (X\ n)) \longrightarrow_{NS} 0 \longleftrightarrow X \longrightarrow_{NS} 0$   
 ⟨*proof*⟩

**lemma** *NSLIMSEQ-rabs-zero*:  $(\lambda n. |f\ n|) \longrightarrow_{NS} 0 \longleftrightarrow f \longrightarrow_{NS} (0::\text{real})$   
 ⟨*proof*⟩

Generalization to other limits.

**lemma** *NSLIMSEQ-imp-rabs*:  $f \longrightarrow_{NS} l \implies (\lambda n. |f\ n|) \longrightarrow_{NS} |l|$   
**for**  $l :: \text{real}$   
 ⟨*proof*⟩

**lemma** *NSLIMSEQ-inverse-zero*:  $\forall y::\text{real}. \exists N. \forall n \geq N. y < f\ n \implies (\lambda n. \text{inverse } (f\ n)) \longrightarrow_{NS} 0$   
 ⟨proof⟩

**lemma** *NSLIMSEQ-inverse-real-of-nat*:  $(\lambda n. \text{inverse } (\text{real } (\text{Suc } n))) \longrightarrow_{NS} 0$   
 ⟨proof⟩

**lemma** *NSLIMSEQ-inverse-real-of-nat-add*:  $(\lambda n. r + \text{inverse } (\text{real } (\text{Suc } n))) \longrightarrow_{NS} r$   
 ⟨proof⟩

**lemma** *NSLIMSEQ-inverse-real-of-nat-add-minus*:  $(\lambda n. r + - \text{inverse } (\text{real } (\text{Suc } n))) \longrightarrow_{NS} r$   
 ⟨proof⟩

**lemma** *NSLIMSEQ-inverse-real-of-nat-add-minus-mult*:  
 $(\lambda n. r * (1 + - \text{inverse } (\text{real } (\text{Suc } n)))) \longrightarrow_{NS} r$   
 ⟨proof⟩

## 10.2 Convergence

**lemma** *nslimI*:  $X \longrightarrow_{NS} L \implies \text{nslim } X = L$   
 ⟨proof⟩

**lemma** *lim-nslim-iff*:  $\text{lim } X = \text{nslim } X$   
 ⟨proof⟩

**lemma** *NSconvergentD*:  $\text{NSconvergent } X \implies \exists L. X \longrightarrow_{NS} L$   
 ⟨proof⟩

**lemma** *NSconvergentI*:  $X \longrightarrow_{NS} L \implies \text{NSconvergent } X$   
 ⟨proof⟩

**lemma** *convergent-NSconvergent-iff*:  $\text{convergent } X = \text{NSconvergent } X$   
 ⟨proof⟩

**lemma** *NSconvergent-NSLIMSEQ-iff*:  $\text{NSconvergent } X \longleftrightarrow X \longrightarrow_{NS} \text{nslim } X$   
 ⟨proof⟩

## 10.3 Bounded Monotonic Sequences

**lemma** *NSBseqD*:  $\text{NSBseq } X \implies N \in \text{HNatInfinite} \implies (*f* X) N \in \text{HFinite}$   
 ⟨proof⟩

**lemma** *Standard-subset-HFfinite*:  $\text{Standard} \subseteq \text{HFfinite}$   
 ⟨proof⟩

**lemma** *NSBseqD2*:  $\text{NSBseq } X \implies (*f* X) N \in \text{HFfinite}$   
 ⟨proof⟩

**lemma** *NSBseqI*:  $\forall N \in \text{HNatInfinite}. (*f* X) N \in \text{HFinite} \implies \text{NSBseq } X$   
 ⟨proof⟩

The standard definition implies the nonstandard definition.

**lemma** *Bseq-NSBseq*:  $\text{Bseq } X \implies \text{NSBseq } X$   
 ⟨proof⟩

The nonstandard definition implies the standard definition.

**lemma** *SReal-less-omega*:  $r \in \mathbf{R} \implies r < \omega$   
 ⟨proof⟩

**lemma** *NSBseq-Bseq*:  $\text{NSBseq } X \implies \text{Bseq } X$   
 ⟨proof⟩

Equivalence of nonstandard and standard definitions for a bounded sequence.

**lemma** *Bseq-NSBseq-iff*:  $\text{Bseq } X = \text{NSBseq } X$   
 ⟨proof⟩

A convergent sequence is bounded: Boundedness as a necessary condition for convergence. The nonstandard version has no existential, as usual.

**lemma** *NSconvergent-NSBseq*:  $\text{NSconvergent } X \implies \text{NSBseq } X$   
 ⟨proof⟩

Standard Version: easily now proved using equivalence of NS and standard definitions.

**lemma** *convergent-Bseq*:  $\text{convergent } X \implies \text{Bseq } X$   
**for**  $X :: \text{nat} \Rightarrow 'b::\text{real-normed-vector}$   
 ⟨proof⟩

### 10.3.1 Upper Bounds and Lubs of Bounded Sequences

**lemma** *NSBseq-isUb*:  $\text{NSBseq } X \implies \exists U::\text{real}. \text{isUb UNIV } \{x. \exists n. X n = x\} U$   
 ⟨proof⟩

**lemma** *NSBseq-isLub*:  $\text{NSBseq } X \implies \exists U::\text{real}. \text{isLub UNIV } \{x. \exists n. X n = x\} U$   
 ⟨proof⟩

### 10.3.2 A Bounded and Monotonic Sequence Converges

The best of both worlds: Easier to prove this result as a standard theorem and then use equivalence to "transfer" it into the equivalent nonstandard form if needed!

**lemma** *Bmonoseq-NSLIMSEQ*:  $\forall_F k \text{ in sequentially. } X k = X m \implies X \longrightarrow_{NS} X m$   
 ⟨proof⟩

**lemma** *NSBseq-mono-NSconvergent*:  $NSBseq\ X \implies \forall m. \forall n \geq m. X\ m \leq X\ n \implies NSconvergent\ X$   
**for**  $X :: nat \Rightarrow real$   
*<proof>*

## 10.4 Cauchy Sequences

**lemma** *NSCauchyI*:  
 $(\bigwedge M\ N. M \in HNatInfinite \implies N \in HNatInfinite \implies starfun\ X\ M \approx starfun\ X\ N) \implies NSCauchy\ X$   
*<proof>*

**lemma** *NSCauchyD*:  
 $NSCauchy\ X \implies M \in HNatInfinite \implies N \in HNatInfinite \implies starfun\ X\ M \approx starfun\ X\ N$   
*<proof>*

### 10.4.1 Equivalence Between NS and Standard

**lemma** *Cauchy-NSCauchy*:  
**assumes**  $X: Cauchy\ X$   
**shows**  $NSCauchy\ X$   
*<proof>*

**lemma** *NSCauchy-Cauchy*:  
**assumes**  $X: NSCauchy\ X$   
**shows**  $Cauchy\ X$   
*<proof>*

**theorem** *NSCauchy-Cauchy-iff*:  $NSCauchy\ X = Cauchy\ X$   
*<proof>*

### 10.4.2 Cauchy Sequences are Bounded

A Cauchy sequence is bounded – nonstandard version.

**lemma** *NSCauchy-NSBseq*:  $NSCauchy\ X \implies NSBseq\ X$   
*<proof>*

### 10.4.3 Cauchy Sequences are Convergent

Equivalence of Cauchy criterion and convergence: We will prove this using our NS formulation which provides a much easier proof than using the standard definition. We do not need to use properties of subsequences such as boundedness, monotonicity etc... Compare with Harrison’s corresponding proof in HOL which is much longer and more complicated. Of course, we do not have problems which he encountered with guessing the right instantiations for his ‘epsilon-delta’ proof(s) in this case since the NS formulations do not involve existential quantifiers.

**lemma** *NSconvergent-NSCauchy*:  $NSconvergent\ X \implies NSCauchy\ X$   
 ⟨proof⟩

**lemma** *real-NSCauchy-NSconvergent*:  
**fixes**  $X :: nat \Rightarrow real$   
**assumes**  $NSCauchy\ X$  **shows**  $NSconvergent\ X$   
 ⟨proof⟩

**lemma** *NSCauchy-NSconvergent*:  $NSCauchy\ X \implies NSconvergent\ X$   
**for**  $X :: nat \Rightarrow 'a::banach$   
 ⟨proof⟩

**lemma** *NSCauchy-NSconvergent-iff*:  $NSCauchy\ X = NSconvergent\ X$   
**for**  $X :: nat \Rightarrow 'a::banach$   
 ⟨proof⟩

## 10.5 Power Sequences

The sequence  $x^n$  tends to 0 if  $(0::'a) \leq x$  and  $x < (1::'a)$ . Proof will use (NS) Cauchy equivalence for convergence and also fact that bounded and monotonic sequence converges.

We now use NS criterion to bring proof of theorem through.

**lemma** *NSLIMSEQ-realpow-zero*:  
**fixes**  $x :: real$   
**assumes**  $0 \leq x < 1$  **shows**  $(\lambda n. x \wedge n) \longrightarrow_{NS} 0$   
 ⟨proof⟩

**lemma** *NSLIMSEQ-abs-realpow-zero*:  $|c| < 1 \implies (\lambda n. |c| \wedge n) \longrightarrow_{NS} 0$   
**for**  $c :: real$   
 ⟨proof⟩

**lemma** *NSLIMSEQ-abs-realpow-zero2*:  $|c| < 1 \implies (\lambda n. c \wedge n) \longrightarrow_{NS} 0$   
**for**  $c :: real$   
 ⟨proof⟩

**end**

## 11 Finite Summation and Infinite Series for Hyperreals

**theory** *HSeries*  
**imports** *HSEQ*  
**begin**

**definition**  $sumhr :: hypnat \times hypnat \times (nat \Rightarrow real) \Rightarrow hypreal$   
**where**  $sumhr = (\lambda(M,N,f). starfun2 (\lambda m n. sum\ f\ \{m..<n\})\ M\ N)$



**definition**  $NSsums :: (nat \Rightarrow real) \Rightarrow real \Rightarrow bool$  (**infixr**  $NSsums$  80)  
**where**  $f NSsums s = (\lambda n. sum f \{..<n\}) \longrightarrow_{NS} s$

**definition**  $NSsummable :: (nat \Rightarrow real) \Rightarrow bool$   
**where**  $NSsummable f \longleftrightarrow (\exists s. f NSsums s)$

**definition**  $NSsuminf :: (nat \Rightarrow real) \Rightarrow real$   
**where**  $NSsuminf f = (THE s. f NSsums s)$

**lemma**  $sumhr\text{-app}$ :  $sumhr (M, N, f) = (*f2* (\lambda m n. sum f \{m..<n\})) M N$   
 $\langle proof \rangle$

Base case in definition of  $sumr$ .

**lemma**  $sumhr\text{-zero}$  [ $simp$ ]:  $\bigwedge m. sumhr (m, 0, f) = 0$   
 $\langle proof \rangle$

Recursive case in definition of  $sumr$ .

**lemma**  $sumhr\text{-if}$ :  
 $\bigwedge m n. sumhr (m, n + 1, f) = (if\ n + 1 \leq m\ then\ 0\ else\ sumhr (m, n, f) + (*f* f) n)$   
 $\langle proof \rangle$

**lemma**  $sumhr\text{-Suc-zero}$  [ $simp$ ]:  $\bigwedge n. sumhr (n + 1, n, f) = 0$   
 $\langle proof \rangle$

**lemma**  $sumhr\text{-eq-bounds}$  [ $simp$ ]:  $\bigwedge n. sumhr (n, n, f) = 0$   
 $\langle proof \rangle$

**lemma**  $sumhr\text{-Suc}$  [ $simp$ ]:  $\bigwedge m. sumhr (m, m + 1, f) = (*f* f) m$   
 $\langle proof \rangle$

**lemma**  $sumhr\text{-add-lbound-zero}$  [ $simp$ ]:  $\bigwedge k m. sumhr (m + k, k, f) = 0$   
 $\langle proof \rangle$

**lemma**  $sumhr\text{-add}$ :  $\bigwedge m n. sumhr (m, n, f) + sumhr (m, n, g) = sumhr (m, n, \lambda i. f i + g i)$   
 $\langle proof \rangle$

**lemma**  $sumhr\text{-mult}$ :  $\bigwedge m n. hypreal\text{-of-real } r * sumhr (m, n, f) = sumhr (m, n, \lambda n. r * f n)$   
 $\langle proof \rangle$

**lemma**  $sumhr\text{-split-add}$ :  $\bigwedge n p. n < p \implies sumhr (0, n, f) + sumhr (n, p, f) = sumhr (0, p, f)$   
 $\langle proof \rangle$

**lemma**  $sumhr\text{-split-diff}$ :  $n < p \implies sumhr (0, p, f) - sumhr (0, n, f) = sumhr (n, p, f)$   
 $\langle proof \rangle$

**lemma** *sumhr-hrabs*:  $\bigwedge m n. |\text{sumhr } (m, n, f)| \leq \text{sumhr } (m, n, \lambda i. |f i|)$   
 ⟨proof⟩

Other general version also needed.

**lemma** *sumhr-fun-hypnat-eq*:  
 $(\forall r. m \leq r \wedge r < n \longrightarrow f r = g r) \longrightarrow$   
 $\text{sumhr } (\text{hypnat-of-nat } m, \text{hypnat-of-nat } n, f) =$   
 $\text{sumhr } (\text{hypnat-of-nat } m, \text{hypnat-of-nat } n, g)$   
 ⟨proof⟩

**lemma** *sumhr-const*:  $\bigwedge n. \text{sumhr } (0, n, \lambda i. r) = \text{hypreal-of-hypnat } n * \text{hypreal-of-real } r$   
 ⟨proof⟩

**lemma** *sumhr-less-bounds-zero* [*simp*]:  $\bigwedge m n. n < m \implies \text{sumhr } (m, n, f) = 0$   
 ⟨proof⟩

**lemma** *sumhr-minus*:  $\bigwedge m n. \text{sumhr } (m, n, \lambda i. -f i) = - \text{sumhr } (m, n, f)$   
 ⟨proof⟩

**lemma** *sumhr-shift-bounds*:  
 $\bigwedge m n. \text{sumhr } (m + \text{hypnat-of-nat } k, n + \text{hypnat-of-nat } k, f) =$   
 $\text{sumhr } (m, n, \lambda i. f (i + k))$   
 ⟨proof⟩

### 11.1 Nonstandard Sums

Infinite sums are obtained by summing to some infinite hypernatural (such as *whn*).

**lemma** *sumhr-hypreal-of-hypnat-omega*:  $\text{sumhr } (0, \text{whn}, \lambda i. 1) = \text{hypreal-of-hypnat } \text{whn}$   
 ⟨proof⟩

**lemma** *whn-eq-omega1*:  $\text{hypreal-of-hypnat } \text{whn} = \omega - 1$   
 ⟨proof⟩

**lemma** *sumhr-hypreal-omega-minus-one*:  $\text{sumhr}(0, \text{whn}, \lambda i. 1) = \omega - 1$   
 ⟨proof⟩

**lemma** *sumhr-minus-one-realpow-zero* [*simp*]:  $\bigwedge N. \text{sumhr } (0, N + N, \lambda i. (-1)^\wedge (i + 1)) = 0$   
 ⟨proof⟩

**lemma** *sumhr-interval-const*:  
 $(\forall n. m \leq \text{Suc } n \longrightarrow f n = r) \wedge m \leq na \implies$   
 $\text{sumhr } (\text{hypnat-of-nat } m, \text{hypnat-of-nat } na, f) = \text{hypreal-of-nat } (na - m) * \text{hypreal-of-real } r$

*<proof>*

**lemma** *starfunNat-sumr*:  $\bigwedge N. (*f* (\lambda n. \text{sum } f \{0..<n\})) N = \text{sumhr } (0, N, f)$   
*<proof>*

**lemma** *sumhr-hrabs-approx* [*simp*]:  $\text{sumhr } (0, M, f) \approx \text{sumhr } (0, N, f) \implies |\text{sumhr } (M, N, f)| \approx 0$   
*<proof>*

## 11.2 Infinite sums: Standard and NS theorems

**lemma** *sums-NSsums-iff*:  $f \text{ sums } l \longleftrightarrow f \text{ NSsums } l$   
*<proof>*

**lemma** *summable-NSsummable-iff*:  $\text{summable } f \longleftrightarrow \text{NSsummable } f$   
*<proof>*

**lemma** *suminf-NSsuminf-iff*:  $\text{suminf } f = \text{NSsuminf } f$   
*<proof>*

**lemma** *NSsums-NSsummable*:  $f \text{ NSsums } l \implies \text{NSsummable } f$   
*<proof>*

**lemma** *NSsummable-NSsums*:  $\text{NSsummable } f \implies f \text{ NSsums } (\text{NSsuminf } f)$   
*<proof>*

**lemma** *NSsums-unique*:  $f \text{ NSsums } s \implies s = \text{NSsuminf } f$   
*<proof>*

**lemma** *NSseries-zero*:  $\forall m. n \leq \text{Suc } m \longrightarrow f m = 0 \implies f \text{ NSsums } (\text{sum } f \{..<n\})$   
*<proof>*

**lemma** *NSsummable-NSCauchy*:

$\text{NSsummable } f \longleftrightarrow (\forall M \in \text{HNatInfinite}. \forall N \in \text{HNatInfinite}. |\text{sumhr } (M, N, f)| \approx 0)$  (**is ?L=?R**)  
*<proof>*

Terms of a convergent series tend to zero.

**lemma** *NSsummable-NSLIMSEQ-zero*:  $\text{NSsummable } f \implies f \longrightarrow_{NS} 0$   
*<proof>*

Nonstandard comparison test.

**lemma** *NSsummable-comparison-test*:  $\exists N. \forall n. N \leq n \longrightarrow |f n| \leq g n \implies \text{NSsummable } g \implies \text{NSsummable } f$   
*<proof>*

**lemma** *NSsummable-rabs-comparison-test*:

$\exists N. \forall n. N \leq n \longrightarrow |f n| \leq g n \implies \text{NSsummable } g \implies \text{NSsummable } (\lambda k. |f k|)$   
*<proof>*

end

## 12 Limits and Continuity (Nonstandard)

```
theory HLim
  imports Star
  abbrevs ----> = -□→NS
begin
```

Nonstandard Definitions.

**definition** *NSLIM* :: ('a::real-normed-vector ⇒ 'b::real-normed-vector) ⇒ 'a ⇒ 'b ⇒ bool  
 (((-)/ -(-)/→<sub>NS</sub> (-)) [60, 0, 60] 60)  
**where**  $f \text{ ---} \rightarrow_{NS} L \iff (\forall x. x \neq \text{star-of } a \wedge x \approx \text{star-of } a \longrightarrow (*f* f) x \approx \text{star-of } L)$

**definition** *isNSCont* :: ('a::real-normed-vector ⇒ 'b::real-normed-vector) ⇒ 'a ⇒ bool  
**where** — NS definition dispenses with limit notions  
 $\text{isNSCont } f \ a \iff (\forall y. y \approx \text{star-of } a \longrightarrow (*f* f) y \approx \text{star-of } (f \ a))$

**definition** *isNSUCont* :: ('a::real-normed-vector ⇒ 'b::real-normed-vector) ⇒ bool  
**where**  $\text{isNSUCont } f \iff (\forall x \ y. x \approx y \longrightarrow (*f* f) x \approx (*f* f) y)$

### 12.1 Limits of Functions

**lemma** *NSLIM-I*:  $(\bigwedge x. x \neq \text{star-of } a \implies x \approx \text{star-of } a \implies \text{starfun } f \ x \approx \text{star-of } L) \implies f \text{ ---} \rightarrow_{NS} L$   
 ⟨proof⟩

**lemma** *NSLIM-D*:  $f \text{ ---} \rightarrow_{NS} L \implies x \neq \text{star-of } a \implies x \approx \text{star-of } a \implies \text{starfun } f \ x \approx \text{star-of } L$   
 ⟨proof⟩

Proving properties of limits using nonstandard definition. The properties hold for standard limits as well!

**lemma** *NSLIM-mult*:  $f \text{ ---} \rightarrow_{NS} l \implies g \text{ ---} \rightarrow_{NS} m \implies (\lambda x. f \ x * g \ x) \text{ ---} \rightarrow_{NS} (l * m)$   
**for**  $l \ m :: 'a::\text{real-normed-algebra}$   
 ⟨proof⟩

**lemma** *starfun-scaleR* [*simp*]:  $\text{starfun } (\lambda x. f \ x *_{R} g \ x) = (\lambda x. \text{scaleHR } (\text{starfun } f \ x) (\text{starfun } g \ x))$   
 ⟨proof⟩

**lemma** *NSLIM-scaleR*:  $f \text{ ---} \rightarrow_{NS} l \implies g \text{ ---} \rightarrow_{NS} m \implies (\lambda x. f \ x *_{R} g \ x) \text{ ---} \rightarrow_{NS} (l *_{R} m)$

*<proof>*

**lemma** *NSLIM-add*:  $f -x \rightarrow_{NS} l \implies g -x \rightarrow_{NS} m \implies (\lambda x. f x + g x) -x \rightarrow_{NS} (l + m)$   
*<proof>*

**lemma** *NSLIM-const* [*simp*]:  $(\lambda x. k) -x \rightarrow_{NS} k$   
*<proof>*

**lemma** *NSLIM-minus*:  $f -a \rightarrow_{NS} L \implies (\lambda x. - f x) -a \rightarrow_{NS} -L$   
*<proof>*

**lemma** *NSLIM-diff*:  $f -x \rightarrow_{NS} l \implies g -x \rightarrow_{NS} m \implies (\lambda x. f x - g x) -x \rightarrow_{NS} (l - m)$   
*<proof>*

**lemma** *NSLIM-add-minus*:  $f -x \rightarrow_{NS} l \implies g -x \rightarrow_{NS} m \implies (\lambda x. f x + - g x) -x \rightarrow_{NS} (l + -m)$   
*<proof>*

**lemma** *NSLIM-inverse*:  $f -a \rightarrow_{NS} L \implies L \neq 0 \implies (\lambda x. \text{inverse } (f x)) -a \rightarrow_{NS} (\text{inverse } L)$   
**for**  $L :: 'a::\text{real-normed-div-algebra}$   
*<proof>*

**lemma** *NSLIM-zero*:  
**assumes**  $f: f -a \rightarrow_{NS} l$   
**shows**  $(\lambda x. f(x) - l) -a \rightarrow_{NS} 0$   
*<proof>*

**lemma** *NSLIM-zero-cancel*:  
**assumes**  $(\lambda x. f x - l) -x \rightarrow_{NS} 0$   
**shows**  $f -x \rightarrow_{NS} l$   
*<proof>*

**lemma** *NSLIM-const-eq*:  
**fixes**  $a :: 'a::\text{real-normed-algebra-1}$   
**assumes**  $(\lambda x. k) -a \rightarrow_{NS} l$   
**shows**  $k = l$   
*<proof>*

**lemma** *NSLIM-unique*:  $f -a \rightarrow_{NS} l \implies f -a \rightarrow_{NS} M \implies l = M$   
**for**  $a :: 'a::\text{real-normed-algebra-1}$   
*<proof>*

**lemma** *NSLIM-mult-zero*:  $f -x \rightarrow_{NS} 0 \implies g -x \rightarrow_{NS} 0 \implies (\lambda x. f x * g x) -x \rightarrow_{NS} 0$   
**for**  $f g :: 'a::\text{real-normed-vector} \Rightarrow 'b::\text{real-normed-algebra}$   
*<proof>*

**lemma** *NSLIM-self*:  $(\lambda x. x) -a \rightarrow_{NS} a$   
 ⟨*proof*⟩

### 12.1.1 Equivalence of *filterlim* and *NSLIM*

**lemma** *LIM-NSLIM*:  
**assumes**  $f: f -a \rightarrow L$   
**shows**  $f -a \rightarrow_{NS} L$   
 ⟨*proof*⟩

**lemma** *NSLIM-LIM*:  
**assumes**  $f: f -a \rightarrow_{NS} L$   
**shows**  $f -a \rightarrow L$   
 ⟨*proof*⟩

**theorem** *LIM-NSLIM-iff*:  $f -x \rightarrow L \longleftrightarrow f -x \rightarrow_{NS} L$   
 ⟨*proof*⟩

## 12.2 Continuity

**lemma** *isNSContD*:  $isNSCont f a \implies y \approx \text{star-of } a \implies (*f* f) y \approx \text{star-of } (f a)$   
 ⟨*proof*⟩

**lemma** *isNSCont-NSLIM*:  $isNSCont f a \implies f -a \rightarrow_{NS} (f a)$   
 ⟨*proof*⟩

**lemma** *NSLIM-isNSCont*:  $f -a \rightarrow_{NS} (f a) \implies isNSCont f a$   
 ⟨*proof*⟩

NS continuity can be defined using NS Limit in similar fashion to standard definition of continuity.

**lemma** *isNSCont-NSLIM-iff*:  $isNSCont f a \longleftrightarrow f -a \rightarrow_{NS} (f a)$   
 ⟨*proof*⟩

Hence, NS continuity can be given in terms of standard limit.

**lemma** *isNSCont-LIM-iff*:  $(isNSCont f a) = (f -a \rightarrow (f a))$   
 ⟨*proof*⟩

Moreover, it's trivial now that NS continuity is equivalent to standard continuity.

**lemma** *isNSCont-isCont-iff*:  $isNSCont f a \longleftrightarrow isCont f a$   
 ⟨*proof*⟩

Standard continuity  $\implies$  NS continuity.

**lemma** *isCont-isNSCont*:  $isCont f a \implies isNSCont f a$   
 ⟨*proof*⟩

NS continuity  $\implies$  Standard continuity.

**lemma** *isNSCont-isCont*:  $isNSCont\ f\ a \implies isCont\ f\ a$   
 ⟨proof⟩

Alternative definition of continuity.

Prove equivalence between NS limits – seems easier than using standard definition.

**lemma** *NSLIM-at0-iff*:  $f\ -a \rightarrow_{NS}\ L \iff (\lambda h. f\ (a + h))\ -0 \rightarrow_{NS}\ L$   
 ⟨proof⟩

**lemma** *isNSCont-minus*:  $isNSCont\ f\ a \implies isNSCont\ (\lambda x. -\ f\ x)\ a$   
 ⟨proof⟩

**lemma** *isNSCont-inverse*:  $isNSCont\ f\ x \implies f\ x \neq 0 \implies isNSCont\ (\lambda x. inverse\ (f\ x))\ x$   
**for**  $f :: 'a::real-normed-vector \Rightarrow 'b::real-normed-div-algebra$   
 ⟨proof⟩

**lemma** *isNSCont-const [simp]*:  $isNSCont\ (\lambda x. k)\ a$   
 ⟨proof⟩

**lemma** *isNSCont-abs [simp]*:  $isNSCont\ abs\ a$   
**for**  $a :: real$   
 ⟨proof⟩

### 12.3 Uniform Continuity

**lemma** *isNSUContD*:  $isNSUCont\ f \implies x \approx y \implies (*f* f)\ x \approx (*f* f)\ y$   
 ⟨proof⟩

**lemma** *isUCont-isNSUCont*:  
**fixes**  $f :: 'a::real-normed-vector \Rightarrow 'b::real-normed-vector$   
**assumes**  $f: isUCont\ f$   
**shows**  $isNSUCont\ f$   
 ⟨proof⟩

**lemma** *isNSUCont-isUCont*:  
**fixes**  $f :: 'a::real-normed-vector \Rightarrow 'b::real-normed-vector$   
**assumes**  $f: isNSUCont\ f$   
**shows**  $isUCont\ f$   
 ⟨proof⟩

end

## 13 Differentiation (Nonstandard)

**theory** *HDeriv*  
**imports** *HLim*  
**begin**

Nonstandard Definitions.

**definition** *nsderiv* :: [*a*::*real-normed-field*  $\Rightarrow$  '*a*, '*a*, '*a*]  $\Rightarrow$  *bool*

((*NSDERIV* (-)/ (-)/  $\Rightarrow$  (-)) [1000, 1000, 60] 60)

**where** *NSDERIV* *f* *x*  $\Rightarrow$  *D*  $\longleftrightarrow$

( $\forall h \in \text{Infinitesimal} - \{0\}. (( *f* f)(\text{star-of } x + h) - \text{star-of } (f x)) / h \approx \text{star-of } D$ )

**definition** *NSdifferentiable* :: [*a*::*real-normed-field*  $\Rightarrow$  '*a*, '*a*]  $\Rightarrow$  *bool*

(**infixl** *NSdifferentiable* 60)

**where** *f* *NSdifferentiable* *x*  $\longleftrightarrow (\exists D. \text{NSDERIV } f x \Rightarrow D)$

**definition** *increment* :: (*real*  $\Rightarrow$  *real*)  $\Rightarrow$  *real*  $\Rightarrow$  *hypreal*  $\Rightarrow$  *hypreal*

**where** *increment* *f* *x* *h* =

(*SOME inc. f* *NSdifferentiable* *x*  $\wedge$  *inc* = ( *\*f\* f* ) (*hypreal-of-real* *x* + *h*) - *hypreal-of-real* (*f* *x*))

### 13.1 Derivatives

**lemma** *DERIV-NS-iff*: (*DERIV* *f* *x*  $\Rightarrow$  *D*)  $\longleftrightarrow (\lambda h. (f (x + h) - f x) / h) -0 \rightarrow_{NS} D$

*<proof>*

**lemma** *NS-DERIV-D*: *DERIV* *f* *x*  $\Rightarrow$  *D*  $\implies (\lambda h. (f (x + h) - f x) / h) -0 \rightarrow_{NS} D$

*<proof>*

**lemma** *Infinitesimal-of-hypreal*:

*x*  $\in$  *Infinitesimal*  $\implies (( *f* \text{ of-real } x)::'a::\text{real-normed-div-algebra star}) \in \text{Infinitesimal}$

*<proof>*

**lemma** *of-hypreal-eq-0-iff*:  $\bigwedge x. (( *f* \text{ of-real } x = (0::'a::\text{real-algebra-1 star})) = (x = 0))$

*<proof>*

**lemma** *NSDeriv-unique*:

**assumes** *NSDERIV* *f* *x*  $\Rightarrow$  *D* *NSDERIV* *f* *x*  $\Rightarrow$  *E*

**shows** *NSDERIV* *f* *x*  $\Rightarrow$  *D*  $\implies$  *NSDERIV* *f* *x*  $\Rightarrow$  *E*  $\implies$  *D* = *E*

*<proof>*

First *NSDERIV* in terms of *NSLIM*.

First equivalence.

**lemma** *NSDERIV-NSLIM-iff*: (*NSDERIV* *f* *x*  $\Rightarrow$  *D*)  $\longleftrightarrow (\lambda h. (f (x + h) - f x) / h) -0 \rightarrow_{NS} D$

*<proof>*

Second equivalence.



**lemma** *NSDERIV-NSLIM-iff2*:  $(NSDERIV f x :=> D) \longleftrightarrow (\lambda z. (f z - f x) / (z - x)) -x \rightarrow_{NS} D$   
 ⟨proof⟩

While we’re at it!

**lemma** *NSDERIV-iff2*:  
 $(NSDERIV f x :=> D) \longleftrightarrow$   
 $(\forall w. w \neq \text{star-of } x \wedge w \approx \text{star-of } x \longrightarrow (*f* (\lambda z. (f z - f x) / (z - x))) w \approx \text{star-of } D)$   
 ⟨proof⟩

**lemma** *NSDERIVD5*:  
 $\llbracket NSDERIV f x :=> D; u \approx \text{hypreal-of-real } x \rrbracket \implies$   
 $(*f* (\lambda z. f z - f x)) u \approx \text{hypreal-of-real } D * (u - \text{hypreal-of-real } x)$   
 ⟨proof⟩

**lemma** *NSDERIVD4*:  
 $\llbracket NSDERIV f x :=> D; h \in \text{Infinitesimal} \rrbracket$   
 $\implies (*f* f)(\text{hypreal-of-real } x + h) - \text{hypreal-of-real } (f x) \approx \text{hypreal-of-real } D * h$   
 ⟨proof⟩

Differentiability implies continuity nice and simple "algebraic" proof.

**lemma** *NSDERIV-isNSCont*:  
**assumes** *NSDERIV f x :=> D* **shows** *isNSCont f x*  
 ⟨proof⟩

Differentiation rules for combinations of functions follow from clear, straight-forward, algebraic manipulations.

Constant function.

**lemma** *NSDERIV-const [simp]*: *NSDERIV*  $(\lambda x. k) x :=> 0$   
 ⟨proof⟩

Sum of functions- proved easily.

**lemma** *NSDERIV-add*:  
**assumes** *NSDERIV f x :=> Da* *NSDERIV g x :=> Db*  
**shows** *NSDERIV*  $(\lambda x. f x + g x) x :=> Da + Db$   
 ⟨proof⟩

Product of functions - Proof is simple.

**lemma** *NSDERIV-mult*:  
**assumes** *NSDERIV g x :=> Db* *NSDERIV f x :=> Da*  
**shows** *NSDERIV*  $(\lambda x. f x * g x) x :=> (Da * g x) + (Db * f x)$   
 ⟨proof⟩

Multiplying by a constant.

**lemma** *NSDERIV-cmult*: *NSDERIV f x :=> D*  $\implies$  *NSDERIV*  $(\lambda x. c * f x) x :=> c * D$

*<proof>*

Negation of function.

**lemma** *NSDERIV-minus*:  $NSDERIV f x :> D \implies NSDERIV (\lambda x. - f x) x :> -D$

*<proof>*

Subtraction.

**lemma** *NSDERIV-add-minus*:

$NSDERIV f x :> Da \implies NSDERIV g x :> Db \implies NSDERIV (\lambda x. f x + - g x) x :> Da + - Db$

*<proof>*

**lemma** *NSDERIV-diff*:

$NSDERIV f x :> Da \implies NSDERIV g x :> Db \implies NSDERIV (\lambda x. f x - g x) x :> Da - Db$

*<proof>*

Similarly to the above, the chain rule admits an entirely straightforward derivation. Compare this with Harrison’s HOL proof of the chain rule, which proved to be trickier and required an alternative characterisation of differentiability- the so-called Carathedory derivative. Our main problem is manipulation of terms.

## 13.2 Lemmas

**lemma** *NSDERIV-zero*:

$\llbracket NSDERIV g x :> D; (*f* g) (star-of x + y) = star-of (g x); y \in Infinitesimal; y \neq 0 \rrbracket$

$\implies D = 0$

*<proof>*

Can be proved differently using *NSLIM-isCont-iff*.

**lemma** *NSDERIV-approx*:

$NSDERIV f x :> D \implies h \in Infinitesimal \implies h \neq 0 \implies$

$(*f* f) (star-of x + h) - star-of (f x) \approx 0$

*<proof>*

From one version of differentiability

$f x - f a \text{ ----- } \approx Db x - a$

**lemma** *NSDERIVD1*:

$\llbracket NSDERIV f (g x) :> Da;$

$(*f* g) (star-of x + y) \neq star-of (g x);$

$(*f* g) (star-of x + y) \approx star-of (g x) \rrbracket$

$\implies ((*f* f) ((*f* g) (star-of x + y)) - star-of (f (g x))) / ((*f* g) (star-of x + y) - star-of (g x)) \approx star-of Da$

*<proof>*

From other version of differentiability

$$f(x+h) - f x \text{ -----} \approx Db h$$

**lemma** *NSDERIVD2*:  $[[ \text{NSDERIV } g \ x \ :> \ Db; y \in \text{Infinitesimal}; y \neq 0 ]]$   
 $\implies (( *f* g) (\text{star-of}(x) + y) - \text{star-of}(g \ x)) / y$   
 $\approx \text{star-of}(Db)$

*<proof>*

This proof uses both definitions of differentiability.

**lemma** *NSDERIV-chain*:

$\text{NSDERIV } f \ (g \ x) \ :> \ Da \implies \text{NSDERIV } g \ x \ :> \ Db \implies \text{NSDERIV } (f \circ g) \ x \ :> \ Da * Db$

*<proof>*

Differentiation of natural number powers.

**lemma** *NSDERIV-Id* [*simp*]:  $\text{NSDERIV } (\lambda x. x) \ x \ :> \ 1$

*<proof>*

**lemma** *NSDERIV-cmult-Id* [*simp*]:  $\text{NSDERIV } ((* \ c) \ x) \ :> \ c$

*<proof>*

**lemma** *NSDERIV-inverse*:

**fixes**  $x :: 'a::\text{real-normed-field}$

**assumes**  $x \neq 0$  — can't get rid of  $x \neq (0::'a)$  because it isn't continuous at zero

**shows**  $\text{NSDERIV } (\lambda x. \text{inverse } x) \ x \ :> \ -(\text{inverse } x \wedge \text{Suc } (\text{Suc } 0))$

*<proof>*

### 13.2.1 Equivalence of NS and Standard definitions

**lemma** *divideR-eq-divide*:  $x /_R y = x / y$

*<proof>*

Now equivalence between *NSDERIV* and *DERIV*.

**lemma** *NSDERIV-DERIV-iff*:  $\text{NSDERIV } f \ x \ :> \ D \longleftrightarrow \text{DERIV } f \ x \ :> \ D$

*<proof>*

NS version.

**lemma** *NSDERIV-pow*:  $\text{NSDERIV } (\lambda x. x \wedge n) \ x \ :> \ \text{real } n * (x \wedge (n - \text{Suc } 0))$

*<proof>*

Derivative of inverse.

**lemma** *NSDERIV-inverse-fun*:

$\text{NSDERIV } f \ x \ :> \ d \implies f \ x \neq 0 \implies$

$\text{NSDERIV } (\lambda x. \text{inverse } (f \ x)) \ x \ :> \ -(d * \text{inverse } (f \ x \wedge \text{Suc } (\text{Suc } 0)))$

**for**  $x :: 'a::\{\text{real-normed-field}\}$

*<proof>*

Derivative of quotient.

**lemma** *NSDERIV-quotient*:

**fixes**  $x :: 'a::\text{real-normed-field}$

**shows**  $NSDERIV f x :> d \implies NSDERIV g x :> e \implies g x \neq 0 \implies$

$NSDERIV (\lambda y. f y / g y) x :> (d * g x - (e * f x)) / (g x \wedge Suc (Suc 0))$

*<proof>*

**lemma** *CARAT-NSDERIV*:

$NSDERIV f x :> l \implies \exists g. (\forall z. f z - f x = g z * (z - x)) \wedge isNSCont g x \wedge g x = l$

*<proof>*

**lemma** *hypreal-eq-minus-iff3*:  $x = y + z \longleftrightarrow x + - z = y$

**for**  $x y z :: \text{hypreal}$

*<proof>*

**lemma** *CARAT-DERIVD*:

**assumes** *all*:  $\forall z. f z - f x = g z * (z - x)$

**and** *nsc*:  $isNSCont g x$

**shows**  $NSDERIV f x :> g x$

*<proof>*

### 13.2.2 Differentiability predicate

**lemma** *NSdifferentiableD*:  $f \text{ NSdifferentiable } x \implies \exists D. NSDERIV f x :> D$

*<proof>*

**lemma** *NSdifferentiableI*:  $NSDERIV f x :> D \implies f \text{ NSdifferentiable } x$

*<proof>*

### 13.3 (NS) Increment

**lemma** *incrementI*:

$f \text{ NSdifferentiable } x \implies$

$\text{increment } f x h = (*f* f) (\text{hypreal-of-real } x + h) - \text{hypreal-of-real } (f x)$

*<proof>*

**lemma** *incrementI2*:

$NSDERIV f x :> D \implies$

$\text{increment } f x h = (*f* f) (\text{hypreal-of-real } x + h) - \text{hypreal-of-real } (f x)$

*<proof>*

The Increment theorem – Keisler p. 65.

**lemma** *increment-thm*:

**assumes**  $NSDERIV f x :> D \ h \in \text{Infinitesimal } h \neq 0$

**shows**  $\exists e \in \text{Infinitesimal}. \text{increment } f x h = \text{hypreal-of-real } D * h + e * h$

*<proof>*

**lemma** *increment-approx-zero*:  $NSDERIV f x :> D \implies h \approx 0 \implies h \neq 0 \implies$   
*increment f x h  $\approx 0$*

*<proof>*

**end**

## 14 Nonstandard Extensions of Transcendental Functions

**theory** *HTranscendental*

**imports** *Complex-Main HSeries HDeriv*

**begin**

**definition**

*exp hr* :: *real*  $\Rightarrow$  *hypreal* **where**

— define exponential function using standard part

*exp hr x*  $\equiv$  *st*(*sum hr* (0, *whn*,  $\lambda n.$  *inverse* (*fact n*) \* (*x*  $\wedge$  *n*)))

**definition**

*sin hr* :: *real*  $\Rightarrow$  *hypreal* **where**

*sin hr x*  $\equiv$  *st*(*sum hr* (0, *whn*,  $\lambda n.$  *sin-coeff n* \* *x*  $\wedge$  *n*))

**definition**

*cos hr* :: *real*  $\Rightarrow$  *hypreal* **where**

*cos hr x*  $\equiv$  *st*(*sum hr* (0, *whn*,  $\lambda n.$  *cos-coeff n* \* *x*  $\wedge$  *n*))

### 14.1 Nonstandard Extension of Square Root Function

**lemma** *STAR-sqrt-zero* [*simp*]: (*\*f\* sqrt*) 0 = 0

*<proof>*

**lemma** *STAR-sqrt-one* [*simp*]: (*\*f\* sqrt*) 1 = 1

*<proof>*

**lemma** *hypreal-sqrt-pow2-iff*: ((*\*f\* sqrt*)(*x*)  $\wedge$  2 = *x*) = (0  $\leq$  *x*)

*<proof>*

**lemma** *hypreal-sqrt-gt-zero-pow2*:  $\bigwedge x. 0 < x \implies$  (*\*f\* sqrt*) (*x*)  $\wedge$  2 = *x*

*<proof>*

**lemma** *hypreal-sqrt-pow2-gt-zero*: 0 < *x*  $\implies$  0 < (*\*f\* sqrt*) (*x*)  $\wedge$  2

*<proof>*

**lemma** *hypreal-sqrt-not-zero*: 0 < *x*  $\implies$  (*\*f\* sqrt*) (*x*)  $\neq$  0

*<proof>*

**lemma** *hypreal-inverse-sqrt-pow2*:

0 < *x*  $\implies$  *inverse* ((*\*f\* sqrt*)(*x*)  $\wedge$  2) = *inverse x*

$\langle proof \rangle$

**lemma** *hypreal-sqrt-mult-distrib*:

$$\bigwedge x y. \llbracket 0 < x; 0 < y \rrbracket \implies \\ (*f* \text{ sqrt})(x*y) = (*f* \text{ sqrt})(x) * (*f* \text{ sqrt})(y)$$

$\langle proof \rangle$

**lemma** *hypreal-sqrt-mult-distrib2*:

$$\llbracket 0 \leq x; 0 \leq y \rrbracket \implies (*f* \text{ sqrt})(x*y) = (*f* \text{ sqrt})(x) * (*f* \text{ sqrt})(y)$$

$\langle proof \rangle$

**lemma** *hypreal-sqrt-approx-zero* [simp]:

assumes  $0 < x$

shows  $(( *f* \text{ sqrt} ) x \approx 0) \longleftrightarrow (x \approx 0)$

$\langle proof \rangle$

**lemma** *hypreal-sqrt-approx-zero2* [simp]:

$$0 \leq x \implies (( *f* \text{ sqrt} )(x) \approx 0) = (x \approx 0)$$

$\langle proof \rangle$

**lemma** *hypreal-sqrt-gt-zero*:  $\bigwedge x. 0 < x \implies 0 < (*f* \text{ sqrt})(x)$

$\langle proof \rangle$

**lemma** *hypreal-sqrt-ge-zero*:  $0 \leq x \implies 0 \leq (*f* \text{ sqrt})(x)$

$\langle proof \rangle$

**lemma** *hypreal-sqrt-lessI*:

$$\bigwedge x u. \llbracket 0 < u; x < u^2 \rrbracket \implies (*f* \text{ sqrt} ) x < u$$

$\langle proof \rangle$

**lemma** *hypreal-sqrt-hrabs* [simp]:  $\bigwedge x. (*f* \text{ sqrt} )(x^2) = |x|$

$\langle proof \rangle$

**lemma** *hypreal-sqrt-hrabs2* [simp]:  $\bigwedge x. (*f* \text{ sqrt} )(x*x) = |x|$

$\langle proof \rangle$

**lemma** *hypreal-sqrt-hyperpow-hrabs* [simp]:

$$\bigwedge x. (*f* \text{ sqrt} )(x \text{ pow } (\text{hypnat-of-nat } 2)) = |x|$$

$\langle proof \rangle$

**lemma** *star-sqrt-HFinite*:  $\llbracket x \in HFinite; 0 \leq x \rrbracket \implies (*f* \text{ sqrt} ) x \in HFinite$

$\langle proof \rangle$

**lemma** *st-hypreal-sqrt*:

assumes  $x \in HFinite$   $0 \leq x$

shows  $st(( *f* \text{ sqrt} ) x) = (*f* \text{ sqrt} )(st x)$

$\langle proof \rangle$

**lemma** *hypreal-sqrt-sum-squares-ge1* [simp]:  $\bigwedge x y. x \leq (*f* \text{ sqrt} )(x^2 + y^2)$

*<proof>*

**lemma** *HFfinite-hypreal-sqrt-imp-HFfinite*:

$\llbracket 0 \leq x; (*f* \text{ sqrt}) x \in \text{HFfinite} \rrbracket \implies x \in \text{HFfinite}$

*<proof>*

**lemma** *HFfinite-hypreal-sqrt-iff [simp]*:

$0 \leq x \implies (( *f* \text{ sqrt}) x \in \text{HFfinite}) = (x \in \text{HFfinite})$

*<proof>*

**lemma** *Infinitesimal-hypreal-sqrt*:

$\llbracket 0 \leq x; x \in \text{Infinitesimal} \rrbracket \implies (*f* \text{ sqrt}) x \in \text{Infinitesimal}$

*<proof>*

**lemma** *Infinitesimal-hypreal-sqrt-imp-Infinitesimal*:

$\llbracket 0 \leq x; (*f* \text{ sqrt}) x \in \text{Infinitesimal} \rrbracket \implies x \in \text{Infinitesimal}$

*<proof>*

**lemma** *Infinitesimal-hypreal-sqrt-iff [simp]*:

$0 \leq x \implies (( *f* \text{ sqrt}) x \in \text{Infinitesimal}) = (x \in \text{Infinitesimal})$

*<proof>*

**lemma** *HInfinite-hypreal-sqrt*:

$\llbracket 0 \leq x; x \in \text{HInfinite} \rrbracket \implies (*f* \text{ sqrt}) x \in \text{HInfinite}$

*<proof>*

**lemma** *HInfinite-hypreal-sqrt-imp-HInfinite*:

$\llbracket 0 \leq x; (*f* \text{ sqrt}) x \in \text{HInfinite} \rrbracket \implies x \in \text{HInfinite}$

*<proof>*

**lemma** *HInfinite-hypreal-sqrt-iff [simp]*:

$0 \leq x \implies (( *f* \text{ sqrt}) x \in \text{HInfinite}) = (x \in \text{HInfinite})$

*<proof>*

**lemma** *HFfinite-exp [simp]*:

$\text{sumhr } (0, \text{whn}, \lambda n. \text{inverse } (\text{fact } n) * x \wedge n) \in \text{HFfinite}$

*<proof>*

**lemma** *exp-hr-zero [simp]*:  $\text{exp-hr } 0 = 1$

*<proof>*

**lemma** *cosh-hr-zero [simp]*:  $\text{cosh-hr } 0 = 1$

*<proof>*

**lemma** *STAR-exp-zero-approx-one [simp]*:  $( *f* \text{ exp}) (0::\text{hypreal}) \approx 1$

*<proof>*

**lemma** *STAR-exp-Infinitesimal*:

**assumes**  $x \in \text{Infinitesimal}$  **shows**  $( *f* \text{ exp}) (x::\text{hypreal}) \approx 1$

$\langle proof \rangle$

**lemma** *STAR-exp-epsilon* [simp]:  $(** exp) \varepsilon \approx 1$   
 $\langle proof \rangle$

**lemma** *STAR-exp-add*:

$\bigwedge(x::'a:: \{banach,real-normed-field\} star) y. (** exp)(x + y) = (** exp) x * (** exp) y$   
 $\langle proof \rangle$

**lemma** *exphr-hypreal-of-real-exp-eq*:  $exphr x = hypreal-of-real (exp x)$   
 $\langle proof \rangle$

**lemma** *starfun-exp-ge-add-one-self* [simp]:  $\bigwedge x::hypreal. 0 \leq x \implies (1 + x) \leq (** exp) x$   
 $\langle proof \rangle$

exp maps infinities to infinities

**lemma** *starfun-exp-HInfinite*:

**fixes**  $x :: hypreal$   
**assumes**  $x \in HInfinite$   $0 \leq x$   
**shows**  $(** exp) x \in HInfinite$   
 $\langle proof \rangle$

**lemma** *starfun-exp-minus*:

$\bigwedge x::'a:: \{banach,real-normed-field\} star. (** exp) (-x) = inverse((** exp) x)$   
 $\langle proof \rangle$

exp maps infinitesimals to infinitesimals

**lemma** *starfun-exp-Infinitesimal*:

**fixes**  $x :: hypreal$   
**assumes**  $x \in HInfinite$   $x \leq 0$   
**shows**  $(** exp) x \in Infinitesimal$   
 $\langle proof \rangle$

**lemma** *starfun-exp-gt-one* [simp]:  $\bigwedge x::hypreal. 0 < x \implies 1 < (** exp) x$   
 $\langle proof \rangle$

**abbreviation** *real-ln* ::  $real \Rightarrow real$  **where**

$real-ln \equiv ln$

**lemma** *starfun-ln-exp* [simp]:  $\bigwedge x. (** real-ln) ((** exp) x) = x$   
 $\langle proof \rangle$

**lemma** *starfun-exp-ln-iff* [simp]:  $\bigwedge x. ((** exp)((** real-ln) x) = x) = (0 < x)$   
 $\langle proof \rangle$

**lemma** *starfun-exp-ln-eq*:  $\bigwedge u x. (** exp) u = x \implies (** real-ln) x = u$   
 $\langle proof \rangle$



**lemma** *starfun-ln-less-self* [simp]:  $\bigwedge x. 0 < x \implies (*f* \text{ real-ln}) x < x$   
 ⟨proof⟩

**lemma** *starfun-ln-ge-zero* [simp]:  $\bigwedge x. 1 \leq x \implies 0 \leq (*f* \text{ real-ln}) x$   
 ⟨proof⟩

**lemma** *starfun-ln-gt-zero* [simp]:  $\bigwedge x. 1 < x \implies 0 < (*f* \text{ real-ln}) x$   
 ⟨proof⟩

**lemma** *starfun-ln-not-eq-zero* [simp]:  $\bigwedge x. [0 < x; x \neq 1] \implies (*f* \text{ real-ln}) x \neq 0$   
 ⟨proof⟩

**lemma** *starfun-ln-HFinite*:  $[x \in HFinite; 1 \leq x] \implies (*f* \text{ real-ln}) x \in HFinite$   
 ⟨proof⟩

**lemma** *starfun-ln-inverse*:  $\bigwedge x. 0 < x \implies (*f* \text{ real-ln}) (\text{inverse } x) = -(*f* \text{ ln}) x$   
 ⟨proof⟩

**lemma** *starfun-abs-exp-cancel*:  $\bigwedge x. |(*f* \text{ exp}) (x::\text{hypreal})| = (*f* \text{ exp}) x$   
 ⟨proof⟩

**lemma** *starfun-exp-less-mono*:  $\bigwedge x y::\text{hypreal}. x < y \implies (*f* \text{ exp}) x < (*f* \text{ exp}) y$   
 ⟨proof⟩

**lemma** *starfun-exp-HFinite*:  
 fixes  $x :: \text{hypreal}$   
 assumes  $x \in HFinite$   
 shows  $(*f* \text{ exp}) x \in HFinite$   
 ⟨proof⟩

**lemma** *starfun-exp-add-HFinite-Infinitesimal-approx*:  
 fixes  $x :: \text{hypreal}$   
 shows  $[x \in Infinitesimal; z \in HFinite] \implies (*f* \text{ exp}) (z + x::\text{hypreal}) \approx (*f* \text{ exp}) z$   
 ⟨proof⟩

**lemma** *starfun-ln-HInfinite*:  
 $[x \in HInfinite; 0 < x] \implies (*f* \text{ real-ln}) x \in HInfinite$   
 ⟨proof⟩

**lemma** *starfun-exp-HInfinite-Infinitesimal-disj*:  
 fixes  $x :: \text{hypreal}$   
 shows  $x \in HInfinite \implies (*f* \text{ exp}) x \in HInfinite \vee (*f* \text{ exp}) (x::\text{hypreal}) \in Infinitesimal$   
 ⟨proof⟩

**lemma** *starfun-ln-HFinite-not-Infinitesimal*:

$\llbracket x \in \text{HFinite} - \text{Infinitesimal}; 0 < x \rrbracket \implies (*f* \text{ real-ln}) x \in \text{HFinite}$   
 $\langle \text{proof} \rangle$

**lemma** *starfun-ln-Infinitesimal-HInfinite*:

**assumes**  $x \in \text{Infinitesimal}$   $0 < x$   
**shows**  $(*f* \text{ real-ln}) x \in \text{HInfinite}$   
 $\langle \text{proof} \rangle$

**lemma** *starfun-ln-less-zero*:  $\bigwedge x. \llbracket 0 < x; x < 1 \rrbracket \implies (*f* \text{ real-ln}) x < 0$

$\langle \text{proof} \rangle$

**lemma** *starfun-ln-Infinitesimal-less-zero*:

$\llbracket x \in \text{Infinitesimal}; 0 < x \rrbracket \implies (*f* \text{ real-ln}) x < 0$   
 $\langle \text{proof} \rangle$

**lemma** *starfun-ln-HInfinite-gt-zero*:

$\llbracket x \in \text{HInfinite}; 0 < x \rrbracket \implies 0 < (*f* \text{ real-ln}) x$   
 $\langle \text{proof} \rangle$

**lemma** *HFinite-sin [simp]*:  $\text{sumhr } (0, \text{whn}, \lambda n. \text{sin-coeff } n * x \wedge n) \in \text{HFinite}$

$\langle \text{proof} \rangle$

**lemma** *STAR-sin-zero [simp]*:  $(*f* \text{ sin}) 0 = 0$

$\langle \text{proof} \rangle$

**lemma** *STAR-sin-Infinitesimal [simp]*:

**fixes**  $x :: 'a :: \{\text{real-normed-field}, \text{banach}\} \text{ star}$   
**assumes**  $x \in \text{Infinitesimal}$   
**shows**  $(*f* \text{ sin}) x \approx x$   
 $\langle \text{proof} \rangle$

**lemma** *HFinite-cos [simp]*:  $\text{sumhr } (0, \text{whn}, \lambda n. \text{cos-coeff } n * x \wedge n) \in \text{HFinite}$

$\langle \text{proof} \rangle$

**lemma** *STAR-cos-zero [simp]*:  $(*f* \text{ cos}) 0 = 1$

$\langle \text{proof} \rangle$

**lemma** *STAR-cos-Infinitesimal [simp]*:

**fixes**  $x :: 'a :: \{\text{real-normed-field}, \text{banach}\} \text{ star}$   
**assumes**  $x \in \text{Infinitesimal}$   
**shows**  $(*f* \text{ cos}) x \approx 1$   
 $\langle \text{proof} \rangle$

**lemma** *STAR-tan-zero [simp]*:  $(*f* \text{ tan}) 0 = 0$

$\langle \text{proof} \rangle$

**lemma** *STAR-tan-Infinitesimal* [simp]:

**assumes**  $x \in \text{Infinitesimal}$

**shows**  $(** \tan) x \approx x$

$\langle \text{proof} \rangle$

**lemma** *STAR-sin-cos-Infinitesimal-mult*:

**fixes**  $x :: 'a::\{\text{real-normed-field}, \text{banach}\}$  *star*

**shows**  $x \in \text{Infinitesimal} \implies (** \sin) x * (** \cos) x \approx x$

$\langle \text{proof} \rangle$

**lemma** *HFinite-pi*: *hypreal-of-real pi*  $\in$  *HFinite*

$\langle \text{proof} \rangle$

**lemma** *STAR-sin-Infinitesimal-divide*:

**fixes**  $x :: 'a::\{\text{real-normed-field}, \text{banach}\}$  *star*

**shows**  $\llbracket x \in \text{Infinitesimal}; x \neq 0 \rrbracket \implies (** \sin) x/x \approx 1$

$\langle \text{proof} \rangle$

## 14.2 Proving $\sin * (1/n) \times 1/(1/n) \approx 1$ for $n = \infty$

**lemma** *lemma-sin-pi*:

$n \in \text{HNatInfinite}$

$\implies (** \sin) (\text{inverse} (\text{hypreal-of-hypnat } n)) / (\text{inverse} (\text{hypreal-of-hypnat } n))$

$\approx 1$

$\langle \text{proof} \rangle$

**lemma** *STAR-sin-inverse-HNatInfinite*:

$n \in \text{HNatInfinite}$

$\implies (** \sin) (\text{inverse} (\text{hypreal-of-hypnat } n)) * \text{hypreal-of-hypnat } n \approx 1$

$\langle \text{proof} \rangle$

**lemma** *Infinitesimal-pi-divide-HNatInfinite*:

$N \in \text{HNatInfinite}$

$\implies \text{hypreal-of-real } \pi / (\text{hypreal-of-hypnat } N) \in \text{Infinitesimal}$

$\langle \text{proof} \rangle$

**lemma** *pi-divide-HNatInfinite-not-zero* [simp]:

$N \in \text{HNatInfinite} \implies \text{hypreal-of-real } \pi / (\text{hypreal-of-hypnat } N) \neq 0$

$\langle \text{proof} \rangle$

**lemma** *STAR-sin-pi-divide-HNatInfinite-approx-pi*:

**assumes**  $n \in \text{HNatInfinite}$

**shows**  $(** \sin) (\text{hypreal-of-real } \pi / \text{hypreal-of-hypnat } n) * \text{hypreal-of-hypnat } n$

$\approx$

$\text{hypreal-of-real } \pi$

$\langle \text{proof} \rangle$

**lemma** *STAR-sin-pi-divide-HNatInfinite-approx-pi2*:

$$\begin{aligned}
& n \in \mathit{HNatInfinite} \\
& \implies \mathit{hypreal-of-hypnat} \, n * (*f* \sin) (\mathit{hypreal-of-real} \, \pi / (\mathit{hypreal-of-hypnat} \, n)) \\
& \approx \mathit{hypreal-of-real} \, \pi \\
& \langle \mathit{proof} \rangle
\end{aligned}$$

**lemma** *starfunNat-pi-divide-n-Infinitesimal*:

$$N \in \mathit{HNatInfinite} \implies (*f* (\lambda x. \pi / \mathit{real} \, x)) \, N \in \mathit{Infinitesimal}$$

$\langle \mathit{proof} \rangle$

**lemma** *STAR-sin-pi-divide-n-approx*:

**assumes**  $N \in \mathit{HNatInfinite}$   
**shows**  $(*f* \sin) ((*f* (\lambda x. \pi / \mathit{real} \, x)) \, N) \approx \mathit{hypreal-of-real} \, \pi / (\mathit{hypreal-of-hypnat} \, N)$   
 $\langle \mathit{proof} \rangle$

**lemma** *NSLIMSEQ-sin-pi*:  $(\lambda n. \mathit{real} \, n * \sin (\pi / \mathit{real} \, n)) \longrightarrow_{NS} \pi$   
 $\langle \mathit{proof} \rangle$

**lemma** *NSLIMSEQ-cos-one*:  $(\lambda n. \cos (\pi / \mathit{real} \, n)) \longrightarrow_{NS} 1$   
 $\langle \mathit{proof} \rangle$

**lemma** *NSLIMSEQ-sin-cos-pi*:

$$(\lambda n. \mathit{real} \, n * \sin (\pi / \mathit{real} \, n) * \cos (\pi / \mathit{real} \, n)) \longrightarrow_{NS} \pi$$

$\langle \mathit{proof} \rangle$

A familiar approximation to  $\cos x$  when  $x$  is small

**lemma** *STAR-cos-Infinitesimal-approx*:

**fixes**  $x :: 'a :: \{\mathit{real-normed-field}, \mathit{banach}\} \mathit{star}$   
**shows**  $x \in \mathit{Infinitesimal} \implies (*f* \cos) \, x \approx 1 - x^2$   
 $\langle \mathit{proof} \rangle$

**lemma** *STAR-cos-Infinitesimal-approx2*:

**fixes**  $x :: \mathit{hypreal}$   
**assumes**  $x \in \mathit{Infinitesimal}$   
**shows**  $(*f* \cos) \, x \approx 1 - (x^2)/2$   
 $\langle \mathit{proof} \rangle$

**end**

## 15 Non-Standard Complex Analysis

**theory** *NSCA*

**imports** *NSComplex HTranscendental*

**begin**

**abbreviation**

$$\begin{aligned}
\mathit{SComplex} & :: \mathit{hcomplex} \, \mathit{set} \, \mathbf{where} \\
\mathit{SComplex} & \equiv \mathit{Standard}
\end{aligned}$$

**definition** — standard part map

$stc :: hcomplex \Rightarrow hcomplex$  **where**

$stc\ x = (SOME\ r.\ x \in HFinite \wedge r \in SComplex \wedge r \approx x)$

## 15.1 Closure Laws for SComplex, the Standard Complex Numbers

**lemma** *SComplex-minus-iff* [simp]:  $(-x \in SComplex) = (x \in SComplex)$

*<proof>*

**lemma** *SComplex-add-cancel*:

$\llbracket x + y \in SComplex; y \in SComplex \rrbracket \Longrightarrow x \in SComplex$

*<proof>*

**lemma** *SReal-hcmod-hcomplex-of-complex* [simp]:

$hcmod\ (hcomplex-of-complex\ r) \in \mathbb{R}$

*<proof>*

**lemma** *SReal-hcmod-numeral*:  $hcmod\ (numeral\ w :: hcomplex) \in \mathbb{R}$

*<proof>*

**lemma** *SReal-hcmod-SComplex*:  $x \in SComplex \Longrightarrow hcmod\ x \in \mathbb{R}$

*<proof>*

**lemma** *SComplex-divide-numeral*:

$r \in SComplex \Longrightarrow r / (numeral\ w :: hcomplex) \in SComplex$

*<proof>*

**lemma** *SComplex-UNIV-complex*:

$\{x.\ hcomplex-of-complex\ x \in SComplex\} = (UNIV :: complex\ set)$

*<proof>*

**lemma** *SComplex-iff*:  $(x \in SComplex) = (\exists y.\ x = hcomplex-of-complex\ y)$

*<proof>*

**lemma** *hcomplex-of-complex-image*:

$range\ hcomplex-of-complex = SComplex$

*<proof>*

**lemma** *inv-hcomplex-of-complex-image*:  $inv\ hcomplex-of-complex\ \text{'}SComplex = UNIV$

*<proof>*

**lemma** *SComplex-hcomplex-of-complex-image*:

$\llbracket \exists x.\ x \in P; P \leq SComplex \rrbracket \Longrightarrow \exists Q.\ P = hcomplex-of-complex\ \text{'}\ Q$

*<proof>*

**lemma** *SComplex-SReal-dense*:

$\llbracket x \in SComplex; y \in SComplex; hcmod\ x < hcmod\ y \rrbracket$

$\mathbb{I} \implies \exists r \in \text{Reals. } \text{hcm}od\ x < r \wedge r < \text{hcm}od\ y$   
 ⟨proof⟩

## 15.2 The Finite Elements form a Subring

**lemma** *HFinite-hcm}od-hcomplex-of-complex* [simp]:  
 $\text{hcm}od\ (\text{hcomplex-of-complex } r) \in \text{HFinite}$   
 ⟨proof⟩

**lemma** *HFinite-hcm}od-iff* [simp]:  $\text{hcm}od\ x \in \text{HFinite} \iff x \in \text{HFinite}$   
 ⟨proof⟩

**lemma** *HFinite-bounded-hcm}od*:  
 $\llbracket x \in \text{HFinite}; y \leq \text{hcm}od\ x; 0 \leq y \rrbracket \implies y \in \text{HFinite}$   
 ⟨proof⟩

## 15.3 The Complex Infinitesimals form a Subring

**lemma** *Infinitesimal-hcm}od-iff*:  
 $(z \in \text{Infinitesimal}) = (\text{hcm}od\ z \in \text{Infinitesimal})$   
 ⟨proof⟩

**lemma** *HInfinite-hcm}od-iff*:  $(z \in \text{HInfinite}) = (\text{hcm}od\ z \in \text{HInfinite})$   
 ⟨proof⟩

**lemma** *HFinite-diff-Infinitesimal-hcm}od*:  
 $x \in \text{HFinite} - \text{Infinitesimal} \implies \text{hcm}od\ x \in \text{HFinite} - \text{Infinitesimal}$   
 ⟨proof⟩

**lemma** *hcm}od-less-Infinitesimal*:  
 $\llbracket e \in \text{Infinitesimal}; \text{hcm}od\ x < \text{hcm}od\ e \rrbracket \implies x \in \text{Infinitesimal}$   
 ⟨proof⟩

**lemma** *hcm}od-le-Infinitesimal*:  
 $\llbracket e \in \text{Infinitesimal}; \text{hcm}od\ x \leq \text{hcm}od\ e \rrbracket \implies x \in \text{Infinitesimal}$   
 ⟨proof⟩

## 15.4 The “Infinitely Close” Relation

**lemma** *approx-SComplex-mult-cancel-zero*:  
 $\llbracket a \in \text{SComplex}; a \neq 0; a*x \approx 0 \rrbracket \implies x \approx 0$   
 ⟨proof⟩

**lemma** *approx-mult-SComplex1*:  $\llbracket a \in \text{SComplex}; x \approx 0 \rrbracket \implies x*a \approx 0$   
 ⟨proof⟩

**lemma** *approx-mult-SComplex2*:  $\llbracket a \in \text{SComplex}; x \approx 0 \rrbracket \implies a*x \approx 0$   
 ⟨proof⟩

**lemma** *approx-mult-SComplex-zero-cancel-iff* [simp]:

$\llbracket a \in SComplex; a \neq 0 \rrbracket \implies (a*x \approx 0) = (x \approx 0)$   
 ⟨proof⟩

**lemma** *approx-SComplex-mult-cancel*:

$\llbracket a \in SComplex; a \neq 0; a*w \approx a*z \rrbracket \implies w \approx z$   
 ⟨proof⟩

**lemma** *approx-SComplex-mult-cancel-iff1* [simp]:

$\llbracket a \in SComplex; a \neq 0 \rrbracket \implies (a*w \approx a*z) = (w \approx z)$   
 ⟨proof⟩

**lemma** *approx-hcmod-approx-zero*:  $(x \approx y) = (hcmod (y - x) \approx 0)$

⟨proof⟩

**lemma** *approx-approx-zero-iff*:  $(x \approx 0) = (hcmod x \approx 0)$

⟨proof⟩

**lemma** *approx-minus-zero-cancel-iff* [simp]:  $(-x \approx 0) = (x \approx 0)$

⟨proof⟩

**lemma** *Infinitesimal-hcmod-add-diff*:

$u \approx 0 \implies hcmod(x + u) - hcmod x \in Infinitesimal$   
 ⟨proof⟩

**lemma** *approx-hcmod-add-hcmod*:  $u \approx 0 \implies hcmod(x + u) \approx hcmod x$

⟨proof⟩

## 15.5 Zero is the Only Infinitesimal Complex Number

**lemma** *Infinitesimal-less-SComplex*:

$\llbracket x \in SComplex; y \in Infinitesimal; 0 < hcmod x \rrbracket \implies hcmod y < hcmod x$   
 ⟨proof⟩

**lemma** *SComplex-Int-Infinitesimal-zero*:  $SComplex \text{ Int } Infinitesimal = \{0\}$

⟨proof⟩

**lemma** *SComplex-Infinitesimal-zero*:

$\llbracket x \in SComplex; x \in Infinitesimal \rrbracket \implies x = 0$   
 ⟨proof⟩

**lemma** *SComplex-HFinite-diff-Infinitesimal*:

$\llbracket x \in SComplex; x \neq 0 \rrbracket \implies x \in HFinite - Infinitesimal$   
 ⟨proof⟩

**lemma** *numeral-not-Infinitesimal* [simp]:

$numeral w \neq (0::hcomplex) \implies (numeral w::hcomplex) \notin Infinitesimal$   
 ⟨proof⟩

**lemma** *approx-SComplex-not-zero*:

$$\llbracket y \in SComplex; x \approx y; y \neq 0 \rrbracket \implies x \neq 0$$

*<proof>*

**lemma** *SComplex-approx-iff*:

$$\llbracket x \in SComplex; y \in SComplex \rrbracket \implies (x \approx y) = (x = y)$$

*<proof>*

**lemma** *approx-unique-complex*:

$$\llbracket r \in SComplex; s \in SComplex; r \approx x; s \approx x \rrbracket \implies r = s$$

*<proof>*

## 15.6 Properties of *hRe*, *hIm* and *HComplex*

**lemma** *abs-hRe-le-hcmod*:  $\bigwedge x. |hRe\ x| \leq hcmod\ x$

*<proof>*

**lemma** *abs-hIm-le-hcmod*:  $\bigwedge x. |hIm\ x| \leq hcmod\ x$

*<proof>*

**lemma** *Infinitesimal-hRe*:  $x \in Infinitesimal \implies hRe\ x \in Infinitesimal$

*<proof>*

**lemma** *Infinitesimal-hIm*:  $x \in Infinitesimal \implies hIm\ x \in Infinitesimal$

*<proof>*

**lemma** *Infinitesimal-HComplex*:

**assumes**  $x: x \in Infinitesimal$  **and**  $y: y \in Infinitesimal$

**shows**  $HComplex\ x\ y \in Infinitesimal$

*<proof>*

**lemma** *hcomplex-Infinitesimal-iff*:

$$(x \in Infinitesimal) \longleftrightarrow (hRe\ x \in Infinitesimal \wedge hIm\ x \in Infinitesimal)$$

*<proof>*

**lemma** *hRe-diff [simp]*:  $\bigwedge x\ y. hRe\ (x - y) = hRe\ x - hRe\ y$

*<proof>*

**lemma** *hIm-diff [simp]*:  $\bigwedge x\ y. hIm\ (x - y) = hIm\ x - hIm\ y$

*<proof>*

**lemma** *approx-hRe*:  $x \approx y \implies hRe\ x \approx hRe\ y$

*<proof>*

**lemma** *approx-hIm*:  $x \approx y \implies hIm\ x \approx hIm\ y$

*<proof>*

**lemma** *approx-HComplex*:



$\llbracket a \approx b; c \approx d \rrbracket \implies HComplex\ a\ c \approx HComplex\ b\ d$   
 ⟨proof⟩

**lemma** *hcomplex-approx-iff*:  
 $(x \approx y) = (hRe\ x \approx hRe\ y \wedge hIm\ x \approx hIm\ y)$   
 ⟨proof⟩

**lemma** *HFinite-hRe*:  $x \in HFinite \implies hRe\ x \in HFinite$   
 ⟨proof⟩

**lemma** *HFinite-hIm*:  $x \in HFinite \implies hIm\ x \in HFinite$   
 ⟨proof⟩

**lemma** *HFinite-HComplex*:  
 assumes  $x \in HFinite\ y \in HFinite$   
 shows  $HComplex\ x\ y \in HFinite$   
 ⟨proof⟩

**lemma** *hcomplex-HFinite-iff*:  
 $(x \in HFinite) = (hRe\ x \in HFinite \wedge hIm\ x \in HFinite)$   
 ⟨proof⟩

**lemma** *hcomplex-HInfinite-iff*:  
 $(x \in HInfinite) = (hRe\ x \in HInfinite \vee hIm\ x \in HInfinite)$   
 ⟨proof⟩

**lemma** *hcomplex-of-hypreal-approx-iff* [simp]:  
 $(hcomplex-of-hypreal\ x \approx hcomplex-of-hypreal\ z) = (x \approx z)$   
 ⟨proof⟩

**lemma** *stc-part-Ex*:  
 assumes  $x \in HFinite$   
 shows  $\exists t \in SComplex. x \approx t$   
 ⟨proof⟩

**lemma** *stc-part-Ex1*:  $x \in HFinite \implies \exists! t. t \in SComplex \wedge x \approx t$   
 ⟨proof⟩

## 15.7 Theorems About Monads

**lemma** *monad-zero-hcmod-iff*:  $(x \in monad\ 0) = (hcmod\ x \in monad\ 0)$   
 ⟨proof⟩

## 15.8 Theorems About Standard Part

**lemma** *stc-approx-self*:  $x \in HFinite \implies stc\ x \approx x$   
 ⟨proof⟩

**lemma** *stc-SComplex*:  $x \in HFinite \implies stc\ x \in SComplex$

*<proof>*

**lemma** *stc-HFinite*:  $x \in \mathit{HFinite} \implies \mathit{stc} x \in \mathit{HFinite}$   
*<proof>*

**lemma** *stc-unique*:  $\llbracket y \in \mathit{SComplex}; y \approx x \rrbracket \implies \mathit{stc} x = y$   
*<proof>*

**lemma** *stc-SComplex-eq [simp]*:  $x \in \mathit{SComplex} \implies \mathit{stc} x = x$   
*<proof>*

**lemma** *stc-eq-approx*:  
 $\llbracket x \in \mathit{HFinite}; y \in \mathit{HFinite}; \mathit{stc} x = \mathit{stc} y \rrbracket \implies x \approx y$   
*<proof>*

**lemma** *approx-stc-eq*:  
 $\llbracket x \in \mathit{HFinite}; y \in \mathit{HFinite}; x \approx y \rrbracket \implies \mathit{stc} x = \mathit{stc} y$   
*<proof>*

**lemma** *stc-eq-approx-iff*:  
 $\llbracket x \in \mathit{HFinite}; y \in \mathit{HFinite} \rrbracket \implies (x \approx y) = (\mathit{stc} x = \mathit{stc} y)$   
*<proof>*

**lemma** *stc-Infinitesimal-add-SComplex*:  
 $\llbracket x \in \mathit{SComplex}; e \in \mathit{Infinitesimal} \rrbracket \implies \mathit{stc}(x + e) = x$   
*<proof>*

**lemma** *stc-Infinitesimal-add-SComplex2*:  
 $\llbracket x \in \mathit{SComplex}; e \in \mathit{Infinitesimal} \rrbracket \implies \mathit{stc}(e + x) = x$   
*<proof>*

**lemma** *HFinite-stc-Infinitesimal-add*:  
 $x \in \mathit{HFinite} \implies \exists e \in \mathit{Infinitesimal}. x = \mathit{stc}(x) + e$   
*<proof>*

**lemma** *stc-add*:  
 $\llbracket x \in \mathit{HFinite}; y \in \mathit{HFinite} \rrbracket \implies \mathit{stc} (x + y) = \mathit{stc}(x) + \mathit{stc}(y)$   
*<proof>*

**lemma** *stc-zero*:  $\mathit{stc} 0 = 0$   
*<proof>*

**lemma** *stc-one*:  $\mathit{stc} 1 = 1$   
*<proof>*

**lemma** *stc-minus*:  $y \in \mathit{HFinite} \implies \mathit{stc}(-y) = -\mathit{stc}(y)$   
*<proof>*

**lemma** *stc-diff*:

$\llbracket x \in \mathit{HFinite}; y \in \mathit{HFinite} \rrbracket \implies \mathit{stc} (x - y) = \mathit{stc}(x) - \mathit{stc}(y)$   
 ⟨proof⟩

**lemma** *stc-mult*:

$\llbracket x \in \mathit{HFinite}; y \in \mathit{HFinite} \rrbracket$   
 $\implies \mathit{stc} (x * y) = \mathit{stc}(x) * \mathit{stc}(y)$   
 ⟨proof⟩

**lemma** *stc-Infinitesimal*:  $x \in \mathit{Infinitesimal} \implies \mathit{stc} x = 0$

⟨proof⟩

**lemma** *stc-not-Infinitesimal*:  $\mathit{stc}(x) \neq 0 \implies x \notin \mathit{Infinitesimal}$

⟨proof⟩

**lemma** *stc-inverse*:

$\llbracket x \in \mathit{HFinite}; \mathit{stc} x \neq 0 \rrbracket \implies \mathit{stc}(\mathit{inverse} x) = \mathit{inverse} (\mathit{stc} x)$   
 ⟨proof⟩

**lemma** *stc-divide* [simp]:

$\llbracket x \in \mathit{HFinite}; y \in \mathit{HFinite}; \mathit{stc} y \neq 0 \rrbracket$   
 $\implies \mathit{stc}(x/y) = (\mathit{stc} x) / (\mathit{stc} y)$   
 ⟨proof⟩

**lemma** *stc-idempotent* [simp]:  $x \in \mathit{HFinite} \implies \mathit{stc}(\mathit{stc}(x)) = \mathit{stc}(x)$

⟨proof⟩

**lemma** *HFinite-HFinite-hcomplex-of-hypreal*:

$z \in \mathit{HFinite} \implies \mathit{hcomplex-of-hypreal} z \in \mathit{HFinite}$   
 ⟨proof⟩

**lemma** *SComplex-SReal-hcomplex-of-hypreal*:

$x \in \mathbb{R} \implies \mathit{hcomplex-of-hypreal} x \in \mathit{SComplex}$   
 ⟨proof⟩

**lemma** *stc-hcomplex-of-hypreal*:

$z \in \mathit{HFinite} \implies \mathit{stc}(\mathit{hcomplex-of-hypreal} z) = \mathit{hcomplex-of-hypreal} (\mathit{st} z)$   
 ⟨proof⟩

**lemma** *hmod-stc-eq*:

**assumes**  $x \in \mathit{HFinite}$

**shows**  $\mathit{hmod}(\mathit{stc} x) = \mathit{st}(\mathit{hmod} x)$

⟨proof⟩

**lemma** *Infinitesimal-hcnj-iff* [simp]:

$(\mathit{hcnj} z \in \mathit{Infinitesimal}) \longleftrightarrow (z \in \mathit{Infinitesimal})$

⟨proof⟩

**end**

## 16 Star-transforms in NSA, Extending Sets of Complex Numbers and Complex Functions

**theory** *CStar*  
**imports** *NSCA*  
**begin**

### 16.1 Properties of the \*-Transform Applied to Sets of Reals

**lemma** *STARC-hcomplex-of-complex-Int*:  $^{**} X \cap SComplex = hcomplex\text{-of-complex } ^{'} X$   
 ⟨*proof*⟩

**lemma** *lemma-not-hcomplexA*:  $x \notin hcomplex\text{-of-complex } ^{'} A \implies \forall y \in A. x \neq hcomplex\text{-of-complex } y$   
 ⟨*proof*⟩

### 16.2 Theorems about Nonstandard Extensions of Functions

**lemma** *starfunC-hcpow*:  $\bigwedge Z. (^{*} (\lambda z. z \hat{\ } n)) Z = Z \text{ pow hypnat-of-nat } n$   
 ⟨*proof*⟩

**lemma** *starfunCR-cmod*:  $^{*} cmod = hcmod$   
 ⟨*proof*⟩

### 16.3 Internal Functions - Some Redundancy With $^{*}f^{*}$ Now

**lemma** *starfun-Re*:  $(^{*} (\lambda x. Re (f x))) = (\lambda x. hRe ((^{*} f) x))$   
 ⟨*proof*⟩

**lemma** *starfun-Im*:  $(^{*} (\lambda x. Im (f x))) = (\lambda x. hIm ((^{*} f) x))$   
 ⟨*proof*⟩

**lemma** *starfunC-eq-Re-Im-iff*:  
 $(^{*} f) x = z \iff (^{*} (\lambda x. Re (f x))) x = hRe z \wedge (^{*} (\lambda x. Im (f x))) x = hIm z$   
 ⟨*proof*⟩

**lemma** *starfunC-approx-Re-Im-iff*:  
 $(^{*} f) x \approx z \iff (^{*} (\lambda x. Re (f x))) x \approx hRe z \wedge (^{*} (\lambda x. Im (f x))) x \approx hIm z$   
 ⟨*proof*⟩

**end**

## 17 Limits, Continuity and Differentiation for Complex Functions

**theory** *CLim*

```

imports CStar
begin

```

```

declare epsilon-not-zero [simp]

```

```

lemma lemma-complex-mult-inverse-squared [simp]:  $x \neq 0 \implies x * (\text{inverse } x)^2 =$ 
inverse  $x$ 
for  $x :: \text{complex}$ 
<proof>

```

Changing the quantified variable. Install earlier?

```

lemma all-shift:  $(\forall x :: 'a :: \text{comm-ring-1}. P x) \longleftrightarrow (\forall x. P (x - a))$ 
<proof>

```

### 17.1 Limit of Complex to Complex Function

```

lemma NSLIM-Re:  $f -a \rightarrow_{NS} L \implies (\lambda x. \text{Re } (f x)) -a \rightarrow_{NS} \text{Re } L$ 
<proof>

```

```

lemma NSLIM-Im:  $f -a \rightarrow_{NS} L \implies (\lambda x. \text{Im } (f x)) -a \rightarrow_{NS} \text{Im } L$ 
<proof>

```

```

lemma LIM-Re:  $f -a \rightarrow L \implies (\lambda x. \text{Re } (f x)) -a \rightarrow \text{Re } L$ 
for  $f :: 'a :: \text{real-normed-vector} \Rightarrow \text{complex}$ 
<proof>

```

```

lemma LIM-Im:  $f -a \rightarrow L \implies (\lambda x. \text{Im } (f x)) -a \rightarrow \text{Im } L$ 
for  $f :: 'a :: \text{real-normed-vector} \Rightarrow \text{complex}$ 
<proof>

```

```

lemma LIM-cnj:  $f -a \rightarrow L \implies (\lambda x. \text{cnj } (f x)) -a \rightarrow \text{cnj } L$ 
for  $f :: 'a :: \text{real-normed-vector} \Rightarrow \text{complex}$ 
<proof>

```

```

lemma LIM-cnj-iff:  $((\lambda x. \text{cnj } (f x)) -a \rightarrow \text{cnj } L) \longleftrightarrow f -a \rightarrow L$ 
for  $f :: 'a :: \text{real-normed-vector} \Rightarrow \text{complex}$ 
<proof>

```

```

lemma starfun-norm:  $( *f* (\lambda x. \text{norm } (f x))) = (\lambda x. \text{hnorm } (( *f* f) x))$ 
<proof>

```

```

lemma star-of-Re [simp]:  $\text{star-of } (\text{Re } x) = \text{hRe } (\text{star-of } x)$ 
<proof>

```

```

lemma star-of-Im [simp]:  $\text{star-of } (\text{Im } x) = \text{hIm } (\text{star-of } x)$ 
<proof>

```

Another equivalence result.

**lemma** *NSCLIM-NSCRLIM-iff*:  $f -x \rightarrow_{NS} L \longleftrightarrow (\lambda y. \text{cmod } (f y - L)) -x \rightarrow_{NS} 0$

*<proof>*

Much, much easier standard proof.

**lemma** *CLIM-CRLIM-iff*:  $f -x \rightarrow L \longleftrightarrow (\lambda y. \text{cmod } (f y - L)) -x \rightarrow 0$

**for**  $f :: 'a::\text{real-normed-vector} \Rightarrow \text{complex}$

*<proof>*

So this is nicer nonstandard proof.

**lemma** *NSCLIM-NSCRLIM-iff2*:  $f -x \rightarrow_{NS} L \longleftrightarrow (\lambda y. \text{cmod } (f y - L)) -x \rightarrow_{NS} 0$

*<proof>*

**lemma** *NSLIM-NSCRLIM-Re-Im-iff*:

$f -a \rightarrow_{NS} L \longleftrightarrow (\lambda x. \text{Re } (f x)) -a \rightarrow_{NS} \text{Re } L \wedge (\lambda x. \text{Im } (f x)) -a \rightarrow_{NS} \text{Im } L$

*<proof>*

**lemma** *LIM-CRLIM-Re-Im-iff*:  $f -a \rightarrow L \longleftrightarrow (\lambda x. \text{Re } (f x)) -a \rightarrow \text{Re } L \wedge (\lambda x. \text{Im } (f x)) -a \rightarrow \text{Im } L$

**for**  $f :: 'a::\text{real-normed-vector} \Rightarrow \text{complex}$

*<proof>*

## 17.2 Continuity

**lemma** *NSLIM-isContc-iff*:  $f -a \rightarrow_{NS} f a \longleftrightarrow (\lambda h. f (a + h)) -0 \rightarrow_{NS} f a$

*<proof>*

## 17.3 Functions from Complex to Reals

**lemma** *isNSContCR-cmod [simp]*:  $\text{isNSCont } \text{cmod } a$

*<proof>*

**lemma** *isContCR-cmod [simp]*:  $\text{isCont } \text{cmod } a$

*<proof>*

**lemma** *isCont-Re*:  $\text{isCont } f a \implies \text{isCont } (\lambda x. \text{Re } (f x)) a$

**for**  $f :: 'a::\text{real-normed-vector} \Rightarrow \text{complex}$

*<proof>*

**lemma** *isCont-Im*:  $\text{isCont } f a \implies \text{isCont } (\lambda x. \text{Im } (f x)) a$

**for**  $f :: 'a::\text{real-normed-vector} \Rightarrow \text{complex}$

*<proof>*

## 17.4 Differentiation of Natural Number Powers

**lemma** *CDERIV-pow [simp]*:  $\text{DERIV } (\lambda x. x \wedge n) x :> \text{complex-of-real } (\text{real } n) * (x \wedge (n - \text{Suc } 0))$

*<proof>*

Nonstandard version.

**lemma** *NSCDERIV-pow*:  $NSDERIV (\lambda x. x \wedge n) x \text{ :> } \text{complex-of-real } (real\ n) * (x \wedge (n - 1))$   
 ⟨proof⟩

Can't relax the premise  $x \neq (0::'a)$ : it isn't continuous at zero.

**lemma** *NSCDERIV-inverse*:  $x \neq 0 \implies NSDERIV (\lambda x. inverse\ x) x \text{ :> } - (inverse\ x)^2$   
 for  $x :: complex$   
 ⟨proof⟩

**lemma** *CDERIV-inverse*:  $x \neq 0 \implies DERIV (\lambda x. inverse\ x) x \text{ :> } - (inverse\ x)^2$   
 for  $x :: complex$   
 ⟨proof⟩

## 17.5 Derivative of Reciprocals (Function *inverse*)

**lemma** *CDERIV-inverse-fun*:  
 $DERIV\ f\ x \text{ :> } d \implies f\ x \neq 0 \implies DERIV (\lambda x. inverse\ (f\ x))\ x \text{ :> } - (d * inverse\ ((f\ x)^2))$   
 for  $x :: complex$   
 ⟨proof⟩

**lemma** *NSCDERIV-inverse-fun*:  
 $NSDERIV\ f\ x \text{ :> } d \implies f\ x \neq 0 \implies NSDERIV (\lambda x. inverse\ (f\ x))\ x \text{ :> } - (d * inverse\ ((f\ x)^2))$   
 for  $x :: complex$   
 ⟨proof⟩

## 17.6 Derivative of Quotient

**lemma** *CDERIV-quotient*:  
 $DERIV\ f\ x \text{ :> } d \implies DERIV\ g\ x \text{ :> } e \implies g(x) \neq 0 \implies$   
 $DERIV (\lambda y. f\ y / g\ y)\ x \text{ :> } (d * g\ x - (e * f\ x)) / (g\ x)^2$   
 for  $x :: complex$   
 ⟨proof⟩

**lemma** *NSCDERIV-quotient*:  
 $NSDERIV\ f\ x \text{ :> } d \implies NSDERIV\ g\ x \text{ :> } e \implies g\ x \neq (0::complex) \implies$   
 $NSDERIV (\lambda y. f\ y / g\ y)\ x \text{ :> } (d * g\ x - (e * f\ x)) / (g\ x)^2$   
 ⟨proof⟩

## 17.7 Caratheodory Formulation of Derivative at a Point: Standard Proof

**lemma** *CARAT-CDERIVD*:  
 $(\forall z. f\ z - f\ x = g\ z * (z - x)) \wedge isNSCont\ g\ x \wedge g\ x = l \implies NSDERIV\ f\ x \text{ :> } l$   
 ⟨proof⟩

end

## 18 Logarithms: Non-Standard Version

theory HLog

imports HTranscendental

begin

**definition** *powhr* :: *hypreal*  $\Rightarrow$  *hypreal*  $\Rightarrow$  *hypreal* (**infixr** *powhr* 80)  
**where** [*transfer-unfold*]:  $x \text{ powhr } a = \text{starfun2 } (\text{powr}) \ x \ a$

**definition** *hlog* :: *hypreal*  $\Rightarrow$  *hypreal*  $\Rightarrow$  *hypreal*  
**where** [*transfer-unfold*]:  $\text{hlog } a \ x = \text{starfun2 } \text{log } a \ x$

**lemma** *powhr*:  $(\text{star-n } X) \ \text{powhr} \ (\text{star-n } Y) = \text{star-n } (\lambda n. (X \ n) \ \text{powr} \ (Y \ n))$   
*<proof>*

**lemma** *powhr-one-eq-one* [*simp*]:  $\bigwedge a. 1 \ \text{powhr} \ a = 1$   
*<proof>*

**lemma** *powhr-mult*:  $\bigwedge a \ x \ y. 0 < x \Longrightarrow 0 < y \Longrightarrow (x * y) \ \text{powhr} \ a = (x \ \text{powhr} \ a) * (y \ \text{powhr} \ a)$   
*<proof>*

**lemma** *powhr-gt-zero* [*simp*]:  $\bigwedge a \ x. 0 < x \ \text{powhr} \ a \longleftrightarrow x \neq 0$   
*<proof>*

**lemma** *powhr-not-zero* [*simp*]:  $\bigwedge a \ x. x \ \text{powhr} \ a \neq 0 \longleftrightarrow x \neq 0$   
*<proof>*

**lemma** *powhr-divide*:  $\bigwedge a \ x \ y. 0 \leq x \Longrightarrow 0 \leq y \Longrightarrow (x / y) \ \text{powhr} \ a = (x \ \text{powhr} \ a) / (y \ \text{powhr} \ a)$   
*<proof>*

**lemma** *powhr-add*:  $\bigwedge a \ b \ x. x \ \text{powhr} \ (a + b) = (x \ \text{powhr} \ a) * (x \ \text{powhr} \ b)$   
*<proof>*

**lemma** *powhr-powhr*:  $\bigwedge a \ b \ x. (x \ \text{powhr} \ a) \ \text{powhr} \ b = x \ \text{powhr} \ (a * b)$   
*<proof>*

**lemma** *powhr-powhr-swap*:  $\bigwedge a \ b \ x. (x \ \text{powhr} \ a) \ \text{powhr} \ b = (x \ \text{powhr} \ b) \ \text{powhr} \ a$   
*<proof>*

**lemma** *powhr-minus*:  $\bigwedge a \ x. x \ \text{powhr} \ (- a) = \text{inverse } (x \ \text{powhr} \ a)$   
*<proof>*

**lemma** *powhr-minus-divide*:  $x \ \text{powhr} \ (- a) = 1 / (x \ \text{powhr} \ a)$   
*<proof>*



**lemma** *powhr-less-mono*:  $\bigwedge a b x. a < b \implies 1 < x \implies x \text{ powhr } a < x \text{ powhr } b$   
 ⟨proof⟩

**lemma** *powhr-less-cancel*:  $\bigwedge a b x. x \text{ powhr } a < x \text{ powhr } b \implies 1 < x \implies a < b$   
 ⟨proof⟩

**lemma** *powhr-less-cancel-iff* [simp]:  $1 < x \implies x \text{ powhr } a < x \text{ powhr } b \iff a < b$   
 ⟨proof⟩

**lemma** *powhr-le-cancel-iff* [simp]:  $1 < x \implies x \text{ powhr } a \leq x \text{ powhr } b \iff a \leq b$   
 ⟨proof⟩

**lemma** *hlog*:  $\text{hlog} (\text{star-}n X) (\text{star-}n Y) = \text{star-}n (\lambda n. \text{log} (X n) (Y n))$   
 ⟨proof⟩

**lemma** *hlog-starfun-ln*:  $\bigwedge x. (*f* \text{ ln}) x = \text{hlog} ((*f* \text{ exp}) 1) x$   
 ⟨proof⟩

**lemma** *powhr-hlog-cancel* [simp]:  $\bigwedge a x. 0 < a \implies a \neq 1 \implies 0 < x \implies a \text{ powhr} (\text{hlog } a x) = x$   
 ⟨proof⟩

**lemma** *hlog-powhr-cancel* [simp]:  $\bigwedge a y. 0 < a \implies a \neq 1 \implies \text{hlog } a (a \text{ powhr } y) = y$   
 ⟨proof⟩

**lemma** *hlog-mult*:

$\bigwedge a x y. 0 < a \implies a \neq 1 \implies 0 < x \implies 0 < y \implies \text{hlog } a (x * y) = \text{hlog } a x + \text{hlog } a y$   
 ⟨proof⟩

**lemma** *hlog-as-starfun*:  $\bigwedge a x. 0 < a \implies a \neq 1 \implies \text{hlog } a x = (*f* \text{ ln}) x / (*f* \text{ ln}) a$   
 ⟨proof⟩

**lemma** *hlog-eq-div-starfun-ln-mult-hlog*:

$\bigwedge a b x. 0 < a \implies a \neq 1 \implies 0 < b \implies b \neq 1 \implies 0 < x \implies \text{hlog } a x = ((*f* \text{ ln}) b / (*f* \text{ ln}) a) * \text{hlog } b x$   
 ⟨proof⟩

**lemma** *powhr-as-starfun*:  $\bigwedge a x. x \text{ powhr } a = (\text{if } x = 0 \text{ then } 0 \text{ else } (*f* \text{ exp}) (a * (*f* \text{ real-ln}) x))$   
 ⟨proof⟩

**lemma** *HInfinite-powhr*:

$x \in \text{HInfinite} \implies 0 < x \implies a \in \text{HFinite} - \text{Infinitesimal} \implies 0 < a \implies x \text{ powhr } a \in \text{HInfinite}$   
 ⟨proof⟩

**lemma** *hlog-hrabs-HInfinite-Infinitesimal*:

$x \in HFinite - Infinitesimal \implies a \in HInfinite \implies 0 < a \implies hlog\ a\ |x| \in Infinitesimal$   
 ⟨proof⟩

**lemma** *hlog-HInfinite-as-starfun*:  $a \in HInfinite \implies 0 < a \implies hlog\ a\ x = (*f* ln)\ x / (*f* ln)\ a$   
 ⟨proof⟩

**lemma** *hlog-one* [simp]:  $\bigwedge a. hlog\ a\ 1 = 0$   
 ⟨proof⟩

**lemma** *hlog-eq-one* [simp]:  $\bigwedge a. 0 < a \implies a \neq 1 \implies hlog\ a\ a = 1$   
 ⟨proof⟩

**lemma** *hlog-inverse*:  $0 < a \implies a \neq 1 \implies 0 < x \implies hlog\ a\ (inverse\ x) = -\ hlog\ a\ x$   
 ⟨proof⟩

**lemma** *hlog-divide*:  $0 < a \implies a \neq 1 \implies 0 < x \implies 0 < y \implies hlog\ a\ (x / y) = hlog\ a\ x - hlog\ a\ y$   
 ⟨proof⟩

**lemma** *hlog-less-cancel-iff* [simp]:  
 $\bigwedge a\ x\ y. 1 < a \implies 0 < x \implies 0 < y \implies hlog\ a\ x < hlog\ a\ y \longleftrightarrow x < y$   
 ⟨proof⟩

**lemma** *hlog-le-cancel-iff* [simp]:  $1 < a \implies 0 < x \implies 0 < y \implies hlog\ a\ x \leq hlog\ a\ y \longleftrightarrow x \leq y$   
 ⟨proof⟩

end

**theory** *Hyperreal*  
**imports** *HLog*  
**begin**

end  
**theory** *Hypercomplex*  
**imports** *CLim Hyperreal*  
**begin**

end

**theory** *Nonstandard-Analysis*  
**imports** *Hypercomplex*  
**begin**

**end**