

Isabelle/HOL-NSA — Non-Standard Analysis

January 18, 2026

Contents

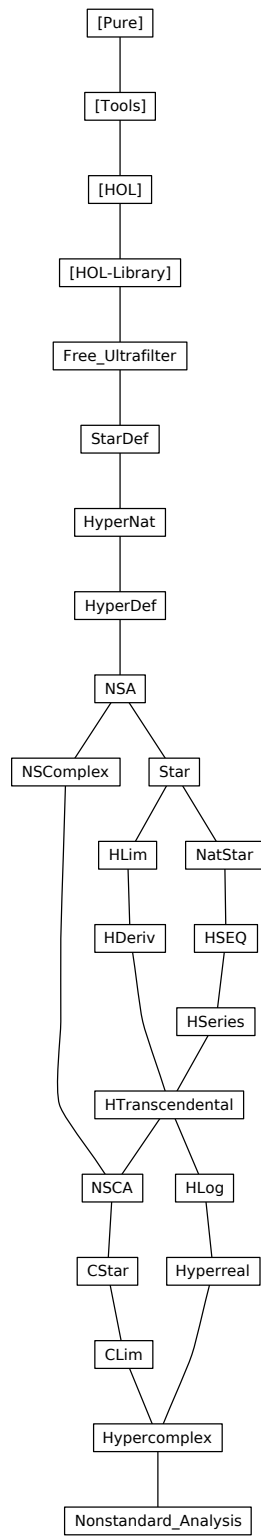
1	Filters and Ultrafilters	3
1.1	Definitions and basic properties	3
1.1.1	Ultrafilters	3
1.2	Maximal filter = Ultrafilter	3
1.3	Ultrafilter Theorem	4
1.3.1	Free Ultrafilters	4
2	Construction of Star Types Using Ultrafilters	5
2.1	A Free Ultrafilter over the Naturals	5
2.2	Definition of <i>star</i> type 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	21
2.13	Number classes	21
2.14	Finite class	22
3	Hypersnatural numbers	22
3.1	Properties Transferred from Naturals	22
3.2	Properties of the set of embedded natural numbers	25
3.3	Infinite Hypersnatural Numbers – <i>HNatInfinite</i>	25
3.3.1	Closure Rules	26
3.4	Existence of an infinite hypersnatural number	26
3.4.1	Alternative characterization of the set of infinite hypersnaturals	27

3.4.2	Alternative Characterization of <i>HNatInfinite</i> using Free Ultrafilter	28
3.5	Embedding of the Hypernaturals into other types	28
4	Construction of Hyperreals Using Ultrafilters	29
4.1	Real vector class instances	30
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>	32
4.5	Properties of <i>star-n</i>	32
4.6	Existence of Infinite Hyperreal Number	33
4.7	Embedding the Naturals into the Hyperreals	33
4.8	Exponentials on the Hyperreals	33
4.9	Powers with Hypernatural Exponents	34
5	Infinite Numbers, Infinitesimals, Infinitely Close Relation	36
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
6	Standard Part Theorem	51
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	62
6.8.1	<i>HFinite</i>	62
6.8.2	<i>HInfinite</i>	62
6.8.3	<i>Infinitesimal</i>	62
6.9	Proof that ω is an infinite number	63
7	Nonstandard Complex Numbers	65
7.0.1	Real and Imaginary parts	65
7.0.2	Imaginary unit	65
7.0.3	Complex conjugate	65
7.0.4	Argand	66
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	68
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	69
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>	74
7.13	Polar Form for Nonstandard Complex Numbers	74
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	77
8	Star-Transforms in Non-Standard Analysis	77
8.1	Preamble - Pulling \exists over \forall	78
8.2	Properties of the Star-transform Applied to Sets of Reals	78
8.3	Theorems about nonstandard extensions of functions	79
9	Star-transforms for the Hypernaturals	82
9.1	Nonstandard Extensions of Functions	83
9.2	Nonstandard Characterization of Induction	85
10	Sequences and Convergence (Nonstandard)	86
10.1	Limits of Sequences	87
10.1.1	Equivalence of <i>LIMSEQ</i> and <i>NSLIMSEQ</i>	88
10.1.2	Derived theorems about <i>NSLIMSEQ</i>	89
10.2	Convergence	89
10.3	Bounded Monotonic Sequences	90
10.3.1	Upper Bounds and Lubs of Bounded Sequences	91
10.3.2	A Bounded and Monotonic Sequence Converges	91
10.4	Cauchy Sequences	91
10.4.1	Equivalence Between NS and Standard	91
10.4.2	Cauchy Sequences are Bounded	92
10.4.3	Cauchy Sequences are Convergent	92
10.5	Power Sequences	92

11 Finite Summation and Infinite Series for Hyperreals	93
11.1 Nonstandard Sums	94
11.2 Infinite sums: Standard and NS theorems	95
12 Limits and Continuity (Nonstandard)	96
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
13 Differentiation (Nonstandard)	100
13.1 Derivatives	100
13.2 Lemmas	102
13.2.1 Equivalence of NS and Standard definitions	104
13.2.2 Differentiability predicate	104
13.3 (NS) Increment	105
14 Nonstandard Extensions of Transcendental Functions	105
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
15 Non-Standard Complex Analysis	113
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	115
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	118
15.8 Theorems About Standard Part	118
16 Star-transforms in NSA, Extending Sets of Complex Numbers and Complex Functions	120
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
17 Limits, Continuity and Differentiation for Complex Functions	121
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	123
17.5 Derivative of Reciprocals (Function <i>inverse</i>)	123
17.6 Derivative of Quotient	124

17.7 Caratheodory Formulation of Derivative at a Point: Standard Proof	124
18 Logarithms: Non-Standard Version	124



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  $F :: 'a \text{ filter}$ 
  assumes proper:  $F \neq \text{bot}$ 
  assumes ultra:  $\text{eventually } P \ F \vee \text{eventually } (\lambda x. \neg P \ x) \ F$ 
begin

```

```

lemma eventually-imp-frequently:  $\text{frequently } P \ F \implies \text{eventually } P \ F$ 
   $\langle \text{proof} \rangle$ 

```

```

lemma frequently-eq-eventually:  $\text{frequently } P \ F = \text{eventually } P \ F$ 
   $\langle \text{proof} \rangle$ 

```

```

lemma eventually-disj-iff:  $\text{eventually } (\lambda x. P \ x \vee Q \ x) \ F \longleftrightarrow \text{eventually } P \ F \vee$ 
 $\text{eventually } Q \ F$ 
   $\langle \text{proof} \rangle$ 

```

```

lemma eventually-all-iff:  $\text{eventually } (\lambda x. \forall y. P \ x \ y) \ F = (\forall Y. \text{eventually } (\lambda x. P$ 
 $x \ (Y \ x)) \ F)$ 
   $\langle \text{proof} \rangle$ 

```

```

lemma eventually-imp-iff:  $\text{eventually } (\lambda x. P \ x \longrightarrow Q \ x) \ F \longleftrightarrow (\text{eventually } P \ F$ 
 $\longrightarrow \text{eventually } Q \ F)$ 
   $\langle \text{proof} \rangle$ 

```

```

lemma eventually-iff-iff:  $\text{eventually } (\lambda x. P \ x \longleftrightarrow Q \ x) \ F \longleftrightarrow (\text{eventually } P \ F$ 
 $\longleftrightarrow \text{eventually } Q \ F)$ 
   $\langle \text{proof} \rangle$ 

```

```

lemma eventually-not-iff:  $\text{eventually } (\lambda x. \neg P \ x) \ F \longleftrightarrow \neg \text{eventually } P \ F$ 
   $\langle \text{proof} \rangle$ 

```

```

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

lemma *max-filter-ultrafilter*:
 assumes $F \neq \text{bot}$
 assumes *max*: $\bigwedge G. G \neq \text{bot} \implies G \leq F \implies F = G$
 shows *ultrafilter* F
 ⟨proof⟩

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

lemma (in *ultrafilter*) *max-filter*:
 assumes $G: G \neq \text{bot}$
 and *sub*: $G \leq F$
 shows $F = G$
 ⟨proof⟩

1.3 Ultrafilter Theorem

lemma *ex-max-ultrafilter*:
 fixes $F :: 'a \ \text{filter}$
 assumes $F: F \neq \text{bot}$
 shows $\exists U \leq F. \text{ultrafilter } U$
 ⟨proof⟩

1.3.1 Free Ultrafilters

There exists a free ultrafilter on any infinite set.

locale *freeultrafilter* = *ultrafilter* +
 assumes *infinite*: $\text{eventually } P \ F \implies \text{infinite } \{x. P \ x\}$
begin

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

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

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

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

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

lemma *ultrafilter*: *ultrafilter* F $\langle \text{proof} \rangle$

end

lemma *freeultrafilter-Ex*:

assumes [*simp*]: *infinite* ($UNIV :: 'a \text{ set}$)

shows $\exists U :: 'a \text{ filter. freeultrafilter } U$

$\langle \text{proof} \rangle$

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}

$\langle \text{proof} \rangle$

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

$\langle \text{proof} \rangle$

2.2 Definition of *star* type constructor

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

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

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

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

$\langle \text{proof} \rangle$

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

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

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

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

$\langle \text{proof} \rangle$

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

$\langle \text{proof} \rangle$

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

Proving that *starrel* is an equivalence relation.

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

lemma *equiv-starrel*: *equiv UNIV starrel*
 $\langle \text{proof} \rangle$

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

lemma *starrel-in-star*: $\text{starrel}^{\{\{x\}\}} \in \text{star}$
 $\langle \text{proof} \rangle$

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

2.3 Transfer principle

This introduction rule starts each transfer proof.

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

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

definition *Ifun* :: $(\text{'a} \Rightarrow \text{'b}) \text{ star} \Rightarrow \text{'a star} \Rightarrow \text{'b star}$
 $(\langle \langle \text{notation} = \langle \text{infix } \star \rangle - \star / - \rangle [300, 301] 300 \rangle$
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* $(\lambda F\ X. \text{starrel}^{\{\{\lambda n. F\ n\ (X\ n)\}\}})$
 $\langle \text{proof} \rangle$

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

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

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

Initialize transfer tactic.

$\langle \text{ML} \rangle$

Transfer introduction rules.

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

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

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

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

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*]:

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

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*]:

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

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

$(\wedge X. f \text{ (star-} n \text{ } X) = g \text{ (star-} n \text{ } X) \equiv \text{eventually } (\lambda n. F \ n \ (X \ n) = G \ n \ (X \ n))$
 $\mathcal{U}) \implies$
 $f = g \equiv \text{eventually } (\lambda n. F \ n = G \ n) \mathcal{U}$
 $\langle \text{proof} \rangle$

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

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

2.4 Standard elements

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

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

$\langle ML \rangle$

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

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

2.5 Internal functions

Transfer tactic should remove occurrences of *Ifun*.

$\langle ML \rangle$

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

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

Nonstandard extensions of functions.

definition *starfun* :: ('a \Rightarrow 'b) \Rightarrow 'a star \Rightarrow 'b star
 $(\langle \langle \text{open-block notation} = \langle \text{prefix starfun} \rangle \rangle * f * - \rangle \ [80] \ 80)$
where $\text{starfun } f \equiv \lambda x. \text{star-of } f \star x$

definition *starfun2* :: ('a \Rightarrow 'b \Rightarrow 'c) \Rightarrow 'a star \Rightarrow 'b star \Rightarrow 'c star
 $(\langle \langle \text{open-block notation} = \langle \text{prefix starfun2} \rangle \rangle * f2 * - \rangle \ [80] \ 80)$
where $\text{starfun2 } f \equiv \lambda x \ y. \text{star-of } f \star x \star y$

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

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

lemma *starfun2-star-n*: $(\ast f2 \ast f) (\text{star-n } X) (\text{star-n } Y) = \text{star-n } (\lambda n. f (X \ n) (Y \ n))$
 $\langle \text{proof} \rangle$

lemma *starfun-star-of [simp]*: $(\ast f \ast f) (\text{star-of } x) = \text{star-of } (f \ x)$
 $\langle \text{proof} \rangle$

lemma *starfun2-star-of [simp]*: $(\ast f2 \ast f) (\text{star-of } x) = \ast f \ast f \ x$
 $\langle \text{proof} \rangle$

lemma *Standard-starfun [simp]*: $x \in \text{Standard} \implies \text{starfun } f \ x \in \text{Standard}$
 $\langle \text{proof} \rangle$

lemma *Standard-starfun2 [simp]*: $x \in \text{Standard} \implies y \in \text{Standard} \implies \text{starfun2 } f \ x \ y \in \text{Standard}$
 $\langle \text{proof} \rangle$

lemma *Standard-starfun-iff*:
assumes *inj*: $\bigwedge x \ y. f \ x = f \ y \implies x = y$
shows $\text{starfun } f \ x \in \text{Standard} \longleftrightarrow x \in \text{Standard}$
 $\langle \text{proof} \rangle$

lemma *Standard-starfun2-iff*:
assumes *inj*: $\bigwedge a \ b \ a' \ b'. f \ a \ b = f \ a' \ b' \implies a = a' \wedge b = b'$
shows $\text{starfun2 } f \ x \ y \in \text{Standard} \longleftrightarrow x \in \text{Standard} \wedge y \in \text{Standard}$
 $\langle \text{proof} \rangle$

2.6 Internal predicates

definition *unstar* :: $\text{bool} \rightarrow \text{bool}$
where $\text{unstar } b \longleftrightarrow b = \text{star-of } \text{True}$

lemma *unstar-star-n*: $\text{unstar } (\text{star-n } P) \longleftrightarrow \text{eventually } P \ \mathcal{U}$
 $\langle \text{proof} \rangle$

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

Transfer tactic should remove occurrences of *unstar*.

$\langle \text{ML} \rangle$

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

definition *starP* :: $(\text{'a} \Rightarrow \text{bool}) \Rightarrow \text{'a} \rightarrow \text{bool}$

$\langle \langle \langle \text{open-block notation} = \langle \text{prefix starP} \rangle \rangle *p* - \rangle \rangle [80] 80)$
where $*p* P = (\lambda x. \text{unstar} (\text{star-of } P \star x))$

definition $\text{starP2} :: ('a \Rightarrow 'b \Rightarrow \text{bool}) \Rightarrow 'a \text{ star} \Rightarrow 'b \text{ star} \Rightarrow \text{bool}$
 $\langle \langle \langle \text{open-block notation} = \langle \text{prefix starP2} \rangle \rangle *p2* - \rangle \rangle [80] 80)$
where $*p2* P = (\lambda x y. \text{unstar} (\text{star-of } P \star x \star y))$

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

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

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

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

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

2.7 Internal sets

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

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

Transfer tactic should remove occurrences of Iset .

$\langle \text{ML} \rangle$

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

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

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

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

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

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

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

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

Nonstandard extensions of sets.

definition *starset* :: ‘a set \Rightarrow ‘a star set

(‘(‘open-block notation=‘prefix starset’)*s* -’) [80] 80)

where *starset* $A = \text{Iset } (\text{star-of } A)$

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

lemma *starset-mem*: $\text{star-of } x \in \text{*s* } A \longleftrightarrow x \in A$

$\langle \text{proof} \rangle$

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

$\langle \text{proof} \rangle$

lemma *starset-empty*: $\text{*s* } \{\} = \{\}$

$\langle \text{proof} \rangle$

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

$\langle \text{proof} \rangle$

lemma *starset-Un*: $\text{*s* } (A \cup B) = \text{*s* } A \cup \text{*s* } B$

$\langle \text{proof} \rangle$

lemma *starset-Int*: $\text{*s* } (A \cap B) = \text{*s* } A \cap \text{*s* } B$

$\langle \text{proof} \rangle$

lemma *starset-Compl*: $\text{*s* } \neg A = \neg (\text{*s* } A)$

$\langle \text{proof} \rangle$

lemma *starset-diff*: $\text{*s* } (A - B) = \text{*s* } A - \text{*s* } B$

$\langle \text{proof} \rangle$

lemma *starset-image*: $\text{*s* } (f \text{ ‘ } A) = (\text{*f* } f) \text{ ‘ } (\text{*s* } A)$

$\langle \text{proof} \rangle$

lemma *starset-vimage*: $*s* (f - ' A) = (*f* f) - ' (*s* A)$
 $\langle proof \rangle$

lemma *starset-subset*: $(*s* A \subseteq *s* B) \longleftrightarrow A \subseteq B$
 $\langle proof \rangle$

lemma *starset-eq*: $(*s* A = *s* B) \longleftrightarrow A = B$
 $\langle proof \rangle$

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 $\langle proof \rangle$
end

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

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

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

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


```

instantiation star :: (minus) minus
begin
  definition star-diff-def:  $(-) \equiv *f2* (-)$ 
  instance  $\langle proof \rangle$ 
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*: $x \in \text{Standard} \implies y \in \text{Standard} \implies x \bmod y \in \text{Standard}$
 ⟨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*: $\text{star-of } (x * y) = \text{star-of } x * \text{star-of } y$
 $\langle \text{proof} \rangle$

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

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

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

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

star-of preserves numerals.

lemma *star-of-zero*: $\text{star-of } 0 = 0$
 $\langle \text{proof} \rangle$

lemma *star-of-one*: $\text{star-of } 1 = 1$
 $\langle \text{proof} \rangle$

star-of preserves orderings.

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

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

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

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*]: *inf* \equiv *f2* *inf*
instance ⟨*proof*⟩
end

instantiation *star* :: (*semilattice-sup*) *semilattice-sup*
begin
definition *star-sup-def* [*transfer-unfold*]: *sup* \equiv *f2* *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*]: $\max = *f2* \max$
 ⟨*proof*⟩

lemma *star-min-def* [*transfer-unfold*]: $\min = *f2* \min$
 ⟨*proof*⟩

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

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

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

lemma *star-of-min* [*simp*]: $\text{star-of } (\min x y) = \min (\text{star-of } x) (\text{star-of } y)$
 ⟨*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]: $(\hat{\cdot}) \equiv \lambda x n. (*f* (\lambda x. x \hat{\cdot} n)) x$
 ⟨proof⟩

lemma *Standard-power* [simp]: $x \in \text{Standard} \implies x \hat{\cdot} n \in \text{Standard}$
 ⟨proof⟩

lemma *star-of-power* [simp]: $\text{star-of } (x \hat{\cdot} n) = \text{star-of } x \hat{\cdot} n$
 ⟨proof⟩

2.13 Number classes

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

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

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

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

lemma *star-of-nat-def* [transfer-unfold]: $\text{of-nat } n = \text{star-of } (\text{of-nat } n)$
 ⟨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]: $\text{of-nat } n \in \text{Standard}$
 ⟨proof⟩

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

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

lemma *Standard-of-int* [*simp*]: *of-int* $z \in \text{Standard}$
 $\langle \text{proof} \rangle$

lemma *star-of-of-int* [*simp*]: *star-of* (*of-int* z) = *of-int* z
 $\langle \text{proof} \rangle$

instance *star* :: (*semiring-char-0*) *semiring-char-0*
 $\langle \text{proof} \rangle$

instance *star* :: (*ring-char-0*) *ring-char-0* $\langle \text{proof} \rangle$

2.14 Finite class

lemma *starset-finite*: *finite* $A \implies *s* A = \text{star-of } A$
 $\langle \text{proof} \rangle$

instance *star* :: (*finite*) *finite*
 $\langle \text{proof} \rangle$

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* $\equiv \text{star-of}$

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

3.1 Properties Transferred from Naturals

lemma *hSuc-not-zero* [*iff*]: $\bigwedge m. \text{hSuc } m \neq 0$
 $\langle \text{proof} \rangle$

lemma *zero-not-hSuc* [*iff*]: $\bigwedge m. 0 \neq \text{hSuc } m$
 $\langle \text{proof} \rangle$

lemma *hSuc-hSuc-eq* [*iff*]: $\bigwedge m n. \text{hSuc } m = \text{hSuc } n \longleftrightarrow m = n$
 $\langle \text{proof} \rangle$

lemma *zero-less-hSuc* [*iff*]: $\bigwedge n. 0 < \text{hSuc } n$
 $\langle \text{proof} \rangle$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

lemma *hypnat-add-diff-inverse*: $\bigwedge m n::\text{hypnat}. \neg m < n \implies n + (m - n) = m$
 ⟨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]: $\bigwedge n::\text{hypnat}. n \neq 0 \longleftrightarrow 0 < n$
 ⟨proof⟩

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

lemma *hypnat-gt-zero-iff2*: $0 < n \longleftrightarrow (\exists m. n = m + 1)$ **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*: $P (a - b) \longleftrightarrow (a < b \longrightarrow P 0) \wedge (\forall d. a = b + d \longrightarrow P d)$
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*: (*Nats* :: *nat star set*) = *Standard*
 ⟨proof⟩

lemma *hypnat-of-nat-mem-Nats* [simp]: *hypnat-of-nat* *n* ∈ *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*::*hypnat*
 shows *of-nat* *m* = *of-nat* *n* + *d* \implies *d* ∈ *range of-nat*
 ⟨proof⟩

lemma *Nats-diff* [simp]: *a* ∈ *Nats* \implies *b* ∈ *Nats* \implies *a* − *b* ∈ *Nats* **for** *a* *b* ::
hypnat
 ⟨proof⟩

3.3 Infinite Hypernatural Numbers – *HNatInfinite*

The set of infinite hypernatural numbers.

definition *HNatInfinite* :: *hypnat set*
 where *HNatInfinite* = {*n*. *n* ∉ *Nats*}

lemma *Nats-not-HNatInfinite-iff*: *x* ∈ *Nats* \longleftrightarrow *x* ∉ *HNatInfinite*
 ⟨proof⟩

lemma *HNatInfinite-not-Nats-iff*: *x* ∈ *HNatInfinite* \longleftrightarrow *x* ∉ *Nats*
 ⟨proof⟩

lemma *star-of-neq-HNatInfinite*: *N* ∈ *HNatInfinite* \implies *star-of* *n* ≠ *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* ∈ *HNatInfinite*
 shows *star-of* *n* < *N*
 ⟨proof⟩

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

3.3.1 Closure Rules

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

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

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

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

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

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

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

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

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

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

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

3.4 Existence of an infinite hypernatural number

ω is in fact an infinite hypernatural number = $[<1, 2, 3, \dots>]$

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

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

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

lemma *whn-not-Nats* [simp]: $whn \notin Nats$
 $\langle proof \rangle$

lemma *HNatInfinite-whn* [simp]: $whn \in HNatInfinite$
 $\langle proof \rangle$

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

lemma *hypnat-of-nat-eq*: $hypnat\text{-}of\text{-}nat\ m = star\text{-}n\ (\lambda n::nat. m)$
 $\langle proof \rangle$

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

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

lemma *Nats-le-whn*: $n \in Nats \implies n \leq whn$
 $\langle proof \rangle$

lemma *hypnat-of-nat-less-whn* [simp]: $hypnat\text{-}of\text{-}nat\ n < whn$
 $\langle proof \rangle$

lemma *hypnat-of-nat-le-whn* [simp]: $hypnat\text{-}of\text{-}nat\ n \leq whn$
 $\langle proof \rangle$

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

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

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 $eventually\ (\lambda n. m < f\ n)\ \mathcal{U}$

$\langle proof \rangle$

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

3.4.2 Alternative Characterization of $HNatInfinite$ using Free Ultrafilter

lemma $HNatInfinite$ -FreeUltrafilterNat:

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

lemma FreeUltrafilterNat- $HNatInfinite$:

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

lemma $HNatInfinite$ -FreeUltrafilterNat-iff:

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

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 $of-hypnat-0\ [simp]: of-hypnat\ 0 = 0$

$\langle proof \rangle$

lemma $of-hypnat-1\ [simp]: of-hypnat\ 1 = 1$

$\langle proof \rangle$

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

$\langle proof \rangle$

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

$\langle proof \rangle$

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

$\langle proof \rangle$

lemma $of-hypnat-less-iff\ [simp]:$

$\bigwedge m\ n. of-hypnat\ m < (of-hypnat\ n :: 'a::linordered-semidom\ star) \longleftrightarrow m < n$
 $\langle proof \rangle$

lemma $of-hypnat-0-less-iff\ [simp]:$

$\bigwedge n. 0 < (of-hypnat\ n :: 'a::linordered-semidom\ star) \longleftrightarrow 0 < n$
 $\langle proof \rangle$

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

$\langle proof \rangle$

lemma $of-hypnat-le-iff\ [simp]:$

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

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

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

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

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

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

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* ($\langle \omega \rangle$)
where $\omega = \text{star-n } (\lambda n. \text{real } (\text{Suc } n))$
— an infinite number = $[<1, 2, 3, \dots>]$

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

4.1 Real vector class instances

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

begin

definition *star-scaleR-def* [*transfer-unfold*]: *scaleR* *r* \equiv *scaleR* *r*

instance \langle *proof* \rangle

end

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

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

instance *star* :: (*real-vector*) *real-vector*
 \langle *proof* \rangle

instance *star* :: (*real-algebra*) *real-algebra*
 \langle *proof* \rangle

instance *star* :: (*real-algebra-1*) *real-algebra-1* \langle *proof* \rangle

instance *star* :: (*real-div-algebra*) *real-div-algebra* \langle *proof* \rangle

instance *star* :: (*field-char-0*) *field-char-0* \langle *proof* \rangle

instance *star* :: (*real-field*) *real-field* \langle *proof* \rangle

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

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

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

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

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

4.2 Injection from hypreal

definition *of-hypreal* :: *hypreal* \Rightarrow '*a*::*real-algebra-1* *star*
 where [*transfer-unfold*]: *of-hypreal* = *scaleR* *of-real*

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

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

lemma *of-hypreal-1* [simp]: *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]:
 $\bigwedge x. \text{of-hypreal } (\text{inverse } x) =$
 $\text{inverse } (\text{of-hypreal } x :: 'a :: \{\text{real-div-algebra}, \text{division-ring}\} \text{ star})$
 ⟨proof⟩

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

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

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

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

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

4.5 Properties of *star-n*

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

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

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

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

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

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

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

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

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

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

lemma *hypreal-omega-gt-zero* [*simp*]: $0 < \omega$
 $\langle \text{proof} \rangle$

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*: *hypreal-of-real* $x \neq \omega$
 $\langle proof \rangle$

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

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

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

lemma *epsilon-not-zero*: $\varepsilon \neq 0$
 $\langle proof \rangle$

lemma *epsilon-inverse-omega*: $\varepsilon = \text{inverse } \omega$
 $\langle proof \rangle$

lemma *epsilon-gt-zero*: $0 < \varepsilon$
 $\langle proof \rangle$

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*: $Nats = \{n. \exists N. n = \text{hypreal-of-nat } N\}$
 $\langle proof \rangle$

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

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

$\langle ML \rangle$

4.8 Exponentials on the Hyperreals

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

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

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

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 $x \ y :: \text{hypreal}$
 $\langle \text{proof} \rangle$

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

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

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

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

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

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* :: $'a::\text{power star} \Rightarrow \text{nat star} \Rightarrow 'a \text{ star}$ (**infixr** $\langle \text{pow} \rangle$ 80)

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

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

$\langle \text{proof} \rangle$

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

$\langle \text{proof} \rangle$

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

$\langle \text{proof} \rangle$

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

$\langle \text{proof} \rangle$

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

$\langle \text{proof} \rangle$

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

$\langle \text{proof} \rangle$

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

$\langle \text{proof} \rangle$

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

$\langle \text{proof} \rangle$

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

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

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

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

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

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

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

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

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

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

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

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

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

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*: $(\text{hypreal-of-real } r)^{\text{pow } (\text{hypnat-of-nat } n)} = \text{hypreal-of-real } (r^{\wedge n})$
 ⟨proof⟩

lemma *hyperpow-SReal [simp]*: $(\text{hypreal-of-real } r)^{\text{pow } (\text{hypnat-of-nat } n)} \in \mathbb{R}$
 ⟨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*: $\bigwedge x. x^{\text{pow } (\text{hypnat-of-nat } n)} = x^{\wedge n}$
 ⟨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* :: $'a::\text{real-normed-vector star} \Rightarrow \text{real star}$
where $[\text{transfer-unfold}]$: $\text{hnorm} = *f* \text{ norm}$

definition *Infinitesimal* :: $('a::\text{real-normed-vector}) \text{ star set}$

where $\text{Infinitesimal} = \{x. \forall r \in \text{Reals}. 0 < r \longrightarrow \text{hnorm } x < r\}$

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

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

definition $\text{approx} :: 'a::\text{real-normed-vector star} \Rightarrow 'a \text{ star} \Rightarrow \text{bool}$ (**infixl** $\langle \approx \rangle$ 50)
where $x \approx y \longleftrightarrow x - y \in \text{Infinitesimal}$
— the “infinitely close” relation

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

definition $\text{monad} :: 'a::\text{real-normed-vector star} \Rightarrow 'a \text{ star set}$
where $\text{monad } x = \{y. x \approx y\}$

definition $\text{galaxy} :: 'a::\text{real-normed-vector star} \Rightarrow 'a \text{ star set}$
where $\text{galaxy } x = \{y. (x + -y) \in \text{HFinite}\}$

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

5.1 Nonstandard Extension of the Norm Function

definition $\text{scaleHR} :: \text{real star} \Rightarrow 'a \text{ star} \Rightarrow 'a::\text{real-normed-vector star}$
where $[\text{transfer-unfold}]: \text{scaleHR} = \text{starfun2 } \text{scaleR}$

lemma $\text{Standard-hnorm} [\text{simp}]: x \in \text{Standard} \Longrightarrow \text{hnorm } x \in \text{Standard}$
 $\langle \text{proof} \rangle$

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

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

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

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

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

$\langle \text{proof} \rangle$

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

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

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

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

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

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

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

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

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

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

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

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

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

a
 $\langle \text{proof} \rangle$

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

lemma *nonzero-hnorm-inverse*:
 $\bigwedge a :: 'a :: \text{real-normed-div-algebra star}. a \neq 0 \implies \text{hnorm } (\text{inverse } a) = \text{inverse } (\text{hnorm } a)$
 $\langle \text{proof} \rangle$

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

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

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

lemma *hnorm-add-less*:
 $\bigwedge (x :: 'a :: \text{real-normed-vector star}) y r s. \text{hnorm } x < r \implies \text{hnorm } y < s \implies \text{hnorm } (x + y) < r + s$
 $\langle \text{proof} \rangle$

lemma *hnorm-mult-less*:
 $\bigwedge (x :: 'a :: \text{real-normed-algebra star}) y r s. \text{hnorm } x < r \implies \text{hnorm } y < s \implies \text{hnorm } (x * y) < r * s$
 $\langle \text{proof} \rangle$

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

5.2 Closure Laws for the Standard Reals

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

lemma *SReal-hrabs*: $x \in \mathbb{R} \implies |x| \in \mathbb{R}$
 for $x :: \text{hypreal}$
 $\langle \text{proof} \rangle$

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

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

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

$\langle \text{proof} \rangle$

lemma *HFinite-hrabs-iff* [iff]: $|x| \in \text{HFinite} \longleftrightarrow x \in \text{HFinite}$
for $x :: \text{hypreal}$
 $\langle \text{proof} \rangle$

lemma *HFinite-hnorm-iff* [iff]: $\text{hnorm } x \in \text{HFinite} \longleftrightarrow x \in \text{HFinite}$
for $x :: \text{hypreal}$
 $\langle \text{proof} \rangle$

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

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

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

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

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

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

5.4 Set of Infinitesimals is a Subring of the Hyperreals

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

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

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

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

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

lemma *Infinitesimal-add*:
assumes $x \in \text{Infinitesimal}$ $y \in \text{Infinitesimal}$

shows $x + y \in \text{Infinitesimal}$
 $\langle \text{proof} \rangle$

lemma *Infinitesimal-minus-iff* [simp]: $- x \in \text{Infinitesimal} \longleftrightarrow x \in \text{Infinitesimal}$
 $\langle \text{proof} \rangle$

lemma *Infinitesimal-hnorm-iff*: $\text{hnorm } x \in \text{Infinitesimal} \longleftrightarrow x \in \text{Infinitesimal}$
 $\langle \text{proof} \rangle$

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

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

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

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

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

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

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

lemma *Infinitesimal-scaleR2*:
assumes $x \in \text{Infinitesimal}$ **shows** $a *_{\mathbb{R}} x \in \text{Infinitesimal}$
 $\langle \text{proof} \rangle$

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

$\langle \text{proof} \rangle$

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

lemma *inverse-Infinitesimal-iff-HInfinite*:
 $x \neq 0 \implies \text{inverse } x \in Infinitesimal \longleftrightarrow x \in HInfinite$
for $x :: 'a::\text{real-normed-div-algebra star}$
 $\langle \text{proof} \rangle$

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

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

lemma *HInfinite-mult*:
fixes $x y :: 'a::\text{real-normed-div-algebra star}$
assumes $x \in HInfinite \ y \in HInfinite$ **shows** $x * y \in HInfinite$
 $\langle \text{proof} \rangle$

lemma *hypreal-add-zero-less-le-mono*: $r < x \implies 0 \leq y \implies r < x + y$
for $r x y :: \text{hypreal}$
 $\langle \text{proof} \rangle$

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

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

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

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

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

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

for $x\ y :: hypreal$
 $\langle proof \rangle$

lemma *not-Infinitesimal-not-zero*: $x \notin Infinitesimal \implies x \neq 0$
 $\langle proof \rangle$

lemma *HFinite-diff-Infinitesimal-hrabs*:
 $x \in HFinite - Infinitesimal \implies |x| \in HFinite - Infinitesimal$
for $x :: hypreal$
 $\langle proof \rangle$

lemma *hnorm-le-Infinitesimal*: $e \in Infinitesimal \implies hnorm\ x \leq e \implies x \in Infinitesimal$
 $\langle proof \rangle$

lemma *hnorm-less-Infinitesimal*: $e \in Infinitesimal \implies hnorm\ x < e \implies x \in Infinitesimal$
 $\langle proof \rangle$

lemma *hrabs-le-Infinitesimal*: $e \in Infinitesimal \implies |x| \leq e \implies x \in Infinitesimal$
for $x :: hypreal$
 $\langle proof \rangle$

lemma *hrabs-less-Infinitesimal*: $e \in Infinitesimal \implies |x| < e \implies x \in Infinitesimal$
for $x :: hypreal$
 $\langle proof \rangle$

lemma *Infinitesimal-interval*:
 $e \in Infinitesimal \implies e' \in Infinitesimal \implies e' < x \implies x < e \implies x \in Infinitesimal$
for $x :: hypreal$
 $\langle proof \rangle$

lemma *Infinitesimal-interval2*:
 $e \in Infinitesimal \implies e' \in Infinitesimal \implies e' \leq x \implies x \leq e \implies x \in Infinitesimal$
for $x :: hypreal$
 $\langle proof \rangle$

lemma *lemma-Infinitesimal-hyperpow*: $x \in Infinitesimal \implies 0 < N \implies |x\ pow\ N| \leq |x|$
for $x :: hypreal$
 $\langle proof \rangle$

lemma *Infinitesimal-hyperpow*: $x \in Infinitesimal \implies 0 < N \implies x\ pow\ N \in Infinitesimal$
for $x :: hypreal$
 $\langle proof \rangle$

lemma *hrealpow-hyperpow-Infinesimal-iff*:

$(x \hat{^} n \in \text{Infinesimal}) \longleftrightarrow x \text{ pow } (\text{hypnat-of-nat } n) \in \text{Infinesimal}$
 $\langle \text{proof} \rangle$

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

for $x :: \text{hypreal}$
 $\langle \text{proof} \rangle$

lemma *not-Infinesimal-mult*:

$x \notin \text{Infinesimal} \implies y \notin \text{Infinesimal} \implies x * y \notin \text{Infinesimal}$
for $x y :: 'a::\text{real-normed-div-algebra star}$
 $\langle \text{proof} \rangle$

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

for $x y :: 'a::\text{real-normed-div-algebra star}$
 $\langle \text{proof} \rangle$

lemma *HFinite-Infinesimal-not-zero*: $x \in \text{HFinite} - \text{Infinesimal} \implies x \neq 0$
 $\langle \text{proof} \rangle$

lemma *HFinite-Infinesimal-diff-mult*:

$x \in \text{HFinite} - \text{Infinesimal} \implies y \in \text{HFinite} - \text{Infinesimal} \implies x * y \in \text{HFinite} - \text{Infinesimal}$
for $x y :: 'a::\text{real-normed-div-algebra star}$
 $\langle \text{proof} \rangle$

lemma *Infinesimal-subset-HFinite*: $\text{Infinesimal} \subseteq \text{HFinite}$
 $\langle \text{proof} \rangle$

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

for $x :: 'a::\text{real-normed-algebra star}$
 $\langle \text{proof} \rangle$

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

for $x :: 'a::\text{real-normed-algebra star}$
 $\langle \text{proof} \rangle$

5.5 The Infinitely Close Relation

lemma *mem-infmal-iff*: $x \in \text{Infinesimal} \longleftrightarrow x \approx 0$
 $\langle \text{proof} \rangle$

lemma *approx-minus-iff*: $x \approx y \longleftrightarrow x - y \approx 0$
 $\langle \text{proof} \rangle$

lemma *approx-minus-iff2*: $x \approx y \longleftrightarrow -y + x \approx 0$
 $\langle \text{proof} \rangle$

lemma *approx-refl [iff]*: $x \approx x$
 $\langle \text{proof} \rangle$

lemma *approx-sym*: $x \approx y \implies y \approx x$
 $\langle \text{proof} \rangle$

lemma *approx-trans*:
 assumes $x \approx y$ $y \approx z$ **shows** $x \approx z$
 $\langle \text{proof} \rangle$

lemma *approx-trans2*: $r \approx x \implies s \approx x \implies r \approx s$
 $\langle \text{proof} \rangle$

lemma *approx-trans3*: $x \approx r \implies x \approx s \implies r \approx s$
 $\langle \text{proof} \rangle$

lemma *approx-reorient*: $x \approx y \longleftrightarrow y \approx x$
 $\langle \text{proof} \rangle$

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

$\langle ML \rangle$

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

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

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

lemma *approx-add*: $a \approx b \implies c \approx d \implies a + c \approx b + d$
 $\langle \text{proof} \rangle$

lemma *approx-minus*: $a \approx b \implies -a \approx -b$
 $\langle \text{proof} \rangle$

lemma *approx-minus2*: $-a \approx -b \implies a \approx b$
 $\langle \text{proof} \rangle$

lemma *approx-minus-cancel [simp]*: $-a \approx -b \longleftrightarrow a \approx b$
 $\langle \text{proof} \rangle$

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

$\langle \text{proof} \rangle$

lemma *approx-diff*: $a \approx b \implies c \approx d \implies a - c \approx b - d$
 $\langle \text{proof} \rangle$

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

lemma *approx-mult2*: $a \approx b \implies c \in HFinite \implies c * a \approx c * b$
for $a \ b \ c :: 'a::\text{real-normed-algebra star}$
 $\langle \text{proof} \rangle$

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

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

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

lemma *approx-eq-imp*: $a = b \implies a \approx b$
 $\langle \text{proof} \rangle$

lemma *Infinitesimal-minus-approx*: $x \in Infinitesimal \implies -x \approx x$
 $\langle \text{proof} \rangle$

lemma *bex-Infinitesimal-iff*: $(\exists y \in Infinitesimal. x - z = y) \longleftrightarrow x \approx z$
 $\langle \text{proof} \rangle$

lemma *bex-Infinitesimal-iff2*: $(\exists y \in Infinitesimal. x = z + y) \longleftrightarrow x \approx z$
 $\langle \text{proof} \rangle$

lemma *Infinitesimal-add-approx*: $y \in Infinitesimal \implies x + y = z \implies x \approx z$
 $\langle \text{proof} \rangle$

lemma *Infinitesimal-add-approx-self*: $y \in Infinitesimal \implies x \approx x + y$
 $\langle \text{proof} \rangle$

lemma *Infinitesimal-add-approx-self2*: $y \in Infinitesimal \implies x \approx y + x$
 $\langle \text{proof} \rangle$

lemma *Infinitesimal-add-minus-approx-self*: $y \in Infinitesimal \implies x \approx x + -y$
 $\langle \text{proof} \rangle$

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

lemma *Infinitesimal-add-right-cancel*: $y \in \text{Infinitesimal} \implies x \approx z + y \implies x \approx z$
 ⟨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*: $x \in \text{HFinite} \implies x \approx y \implies y \in \text{HFinite}$
 ⟨proof⟩

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

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

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

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

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

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

lemma *approx-mult-star-of*: $a \approx \text{star-of } b \implies c \approx \text{star-of } d \implies a * c \approx \text{star-of } b * \text{star-of } d$
for $a \ c :: 'a::\text{real-normed-algebra star}$
 $\langle \text{proof} \rangle$

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 $y :: \text{hypreal}$
 $\langle \text{proof} \rangle$

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

for $y :: \text{hypreal}$
 $\langle \text{proof} \rangle$

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

$\langle \text{proof} \rangle$

lemma *SReal-Infinitesimal-zero*: $x \in \mathbb{R} \implies x \in \text{Infinitesimal} \implies x = 0$

for $x :: \text{hypreal}$
 $\langle \text{proof} \rangle$

lemma *SReal-HFfinite-diff-Infinitesimal*: $x \in \mathbb{R} \implies x \neq 0 \implies x \in \text{HFfinite} - \text{Infinitesimal}$

for $x :: \text{hypreal}$
 $\langle \text{proof} \rangle$

lemma *hypreal-of-real-HFfinite-diff-Infinitesimal*:

$\text{hypreal-of-real } x \neq 0 \implies \text{hypreal-of-real } x \in \text{HFfinite} - \text{Infinitesimal}$
 $\langle \text{proof} \rangle$

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

$\langle \text{proof} \rangle$

lemma *star-of-HFfinite-diff-Infinitesimal*: $x \neq 0 \implies \text{star-of } x \in \text{HFfinite} - \text{Infinitesimal}$

$\langle \text{proof} \rangle$

lemma *numeral-not-Infinitesimal* [simp]:

$\text{numeral } w \neq (0 :: \text{hypreal}) \implies (\text{numeral } w :: \text{hypreal}) \notin \text{Infinitesimal}$
 $\langle \text{proof} \rangle$

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

lemma *one-not-Infinitesimal* [simp]:

$(1 :: 'a :: \{\text{real-normed-vector}, \text{zero-neq-one}\}) \text{ star} \notin \text{Infinitesimal}$
 $\langle \text{proof} \rangle$

lemma *approx-SReal-not-zero*: $y \in \mathbb{R} \implies x \approx y \implies y \neq 0 \implies x \neq 0$

for $x y :: \text{hypreal}$
 $\langle \text{proof} \rangle$

lemma *HFinite-diff-Infinitesimal-approx*:

$x \approx y \implies y \in \text{HFinite} - \text{Infinitesimal} \implies x \in \text{HFinite} - \text{Infinitesimal}$
 $\langle \text{proof} \rangle$

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 $x y :: 'a :: \{\text{real-normed-div-algebra}, \text{field}\} \text{ star}$
 $\langle \text{proof} \rangle$

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

for $x y :: \text{hypreal}$
 $\langle \text{proof} \rangle$

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]: $\text{star-of } x \approx \text{star-of } y \longleftrightarrow x = y$
 $\langle \text{proof} \rangle$

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

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

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

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

lemma [simp]:

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

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

lemma *star-of-approx-zero-iff* [simp]: *star-of* $k \approx 0 \longleftrightarrow k = 0$
 ⟨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 *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 $x \in HFinite$
 shows $\exists t. \text{isLub } \mathbb{R} \ \{s. s \in \mathbb{R} \wedge s < x\} \ t$
 ⟨proof⟩

lemma *hypreal-settle-less-trans*: $S * \leq x \implies x < y \implies S * \leq y$
for $x y :: \text{hypreal}$
 $\langle \text{proof} \rangle$

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

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

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

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

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

Existence of real and Standard Part Theorem.

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

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

There is a unique real infinitely close.

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

6.3 Finite, Infinite and Infinitesimal

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

lemma *HFinite-not-HInfinite*:

assumes $x: x \in \text{HFinite}$ **shows** $x \notin \text{HInfinite}$
 $\langle \text{proof} \rangle$

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

lemma *HInfinite-HFinite-disj*: $x \in \text{HInfinite} \vee x \in \text{HFinite}$
 $\langle \text{proof} \rangle$

lemma *HInfinite-HFinite-iff*: $x \in \text{HInfinite} \longleftrightarrow x \notin \text{HFinite}$
 $\langle \text{proof} \rangle$

lemma *HFinite-HInfinite-iff*: $x \in \text{HFinite} \longleftrightarrow x \notin \text{HInfinite}$
 $\langle \text{proof} \rangle$

lemma *HInfinite-diff-HFinite-Infinitesimal-disj*:

$x \notin \text{Infinitesimal} \implies x \in \text{HInfinite} \vee x \in \text{HFinite} - \text{Infinitesimal}$
 $\langle \text{proof} \rangle$

lemma *HFinite-inverse*: $x \in \text{HFinite} \implies x \notin \text{Infinitesimal} \implies \text{inverse } x \in \text{HFinite}$

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

lemma *HFinite-inverse2*: $x \in \text{HFinite} - \text{Infinitesimal} \implies \text{inverse } x \in \text{HFinite}$

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

Stronger statement possible in fact.

lemma *Infinitesimal-inverse-HFinite*: $x \notin \text{Infinitesimal} \implies \text{inverse } x \in \text{HFinite}$

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

lemma *HFinite-not-Infinitesimal-inverse*:

$x \in \text{HFinite} - \text{Infinitesimal} \implies \text{inverse } x \in \text{HFinite} - \text{Infinitesimal}$
for $x :: 'a::\text{real-normed-div-algebra}$ **star**
 $\langle \text{proof} \rangle$

lemma *approx-inverse*:

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

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

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

lemma *Infinitesimal-square-iff*: $x \in Infinitesimal \longleftrightarrow x * x \in Infinitesimal$
for $x :: 'a::\text{real-normed-div-algebra star}$
 <proof>
declare *Infinitesimal-square-iff* [symmetric, simp]

lemma *HFinite-square-iff* [simp]: $x * x \in HFinite \longleftrightarrow x \in HFinite$
for $x :: 'a::\text{real-normed-div-algebra star}$
 <proof>

lemma *HInfinite-square-iff* [simp]: $x * x \in HInfinite \longleftrightarrow x \in HInfinite$
for $x :: 'a::\text{real-normed-div-algebra star}$
 <proof>

lemma *approx-HFinite-mult-cancel*: $a \in HFinite - Infinitesimal \implies a * w \approx a * z \implies w \approx z$
for $a \ w \ z :: 'a::\text{real-normed-div-algebra star}$
 <proof>

lemma *approx-HFinite-mult-cancel-iff1*: $a \in HFinite - Infinitesimal \implies a * w \approx a * z \longleftrightarrow w \approx z$
for $a \ w \ z :: 'a::\text{real-normed-div-algebra star}$
 <proof>

lemma *HInfinite-HFinite-add-cancel*: $x + y \in HInfinite \implies y \in HFinite \implies x \in HInfinite$
 <proof>

lemma *HInfinite-HFinite-add*: $x \in HInfinite \implies y \in HFinite \implies x + y \in HIn-$

finite
 $\langle \text{proof} \rangle$

lemma *HInfinite-ge-HInfinite*: $x \in HInfinite \implies x \leq y \implies 0 \leq x \implies y \in HInfinite$
for $x y :: \text{hypreal}$
 $\langle \text{proof} \rangle$

lemma *Infinitesimal-inverse-HInfinite*: $x \in Infinitesimal \implies x \neq 0 \implies \text{inverse } x \in HInfinite$
for $x :: 'a::\text{real-normed-div-algebra star}$
 $\langle \text{proof} \rangle$

lemma *HInfinite-HFinite-not-Infinitesimal-mult*:
 $x \in HInfinite \implies y \in HFinite - Infinitesimal \implies x * y \in HInfinite$
for $x y :: 'a::\text{real-normed-div-algebra star}$
 $\langle \text{proof} \rangle$

lemma *HInfinite-HFinite-not-Infinitesimal-mult2*:
 $x \in HInfinite \implies y \in HFinite - Infinitesimal \implies y * x \in HInfinite$
for $x y :: 'a::\text{real-normed-div-algebra star}$
 $\langle \text{proof} \rangle$

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

lemma *HInfinite-gt-zero-gt-one*: $x \in HInfinite \implies 0 < x \implies 1 < x$
for $x :: \text{hypreal}$
 $\langle \text{proof} \rangle$

lemma *not-HInfinite-one [simp]*: $1 \notin HInfinite$
 $\langle \text{proof} \rangle$

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*: $e \in Infinitesimal \implies \text{monad } (x + e) = \text{monad } x$
 $\langle \text{proof} \rangle$

lemma *mem-monad-iff*: $u \in \text{monad } x \longleftrightarrow -u \in \text{monad } (-x)$
 $\langle \text{proof} \rangle$

lemma *Infinitesimal-monad-zero-iff*: $x \in \text{Infinitesimal} \longleftrightarrow x \in \text{monad } 0$
 ⟨proof⟩

lemma *monad-zero-minus-iff*: $x \in \text{monad } 0 \longleftrightarrow -x \in \text{monad } 0$
 ⟨proof⟩

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*: $x < 0 \implies x \notin \text{Infinitesimal} \implies u \in \text{monad } x$

$x \implies u < 0$
for $u \ x :: \text{hypreal}$
 $\langle \text{proof} \rangle$

lemma *lemma-approx-gt-zero*: $0 < x \implies x \notin \text{Infinitesimal} \implies x \approx y \implies 0 < y$
for $x \ y :: \text{hypreal}$
 $\langle \text{proof} \rangle$

lemma *lemma-approx-less-zero*: $x < 0 \implies x \notin \text{Infinitesimal} \implies x \approx y \implies y < 0$
for $x \ y :: \text{hypreal}$
 $\langle \text{proof} \rangle$

lemma *approx-hrabs*: $x \approx y \implies |x| \approx |y|$
for $x \ y :: \text{hypreal}$
 $\langle \text{proof} \rangle$

lemma *approx-hrabs-zero-cancel*: $|x| \approx 0 \implies x \approx 0$
for $x :: \text{hypreal}$
 $\langle \text{proof} \rangle$

lemma *approx-hrabs-add-Infinitesimal*: $e \in \text{Infinitesimal} \implies |x| \approx |x + e|$
for $e \ x :: \text{hypreal}$
 $\langle \text{proof} \rangle$

lemma *approx-hrabs-add-minus-Infinitesimal*: $e \in \text{Infinitesimal} \implies |x| \approx |x + -e|$
for $e \ x :: \text{hypreal}$
 $\langle \text{proof} \rangle$

lemma *hrabs-add-Infinitesimal-cancel*:
 $e \in \text{Infinitesimal} \implies e' \in \text{Infinitesimal} \implies |x + e| = |y + e'| \implies |x| \approx |y|$
for $e \ e' \ x \ y :: \text{hypreal}$
 $\langle \text{proof} \rangle$

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 $e \ e' \ x \ y :: \text{hypreal}$
 $\langle \text{proof} \rangle$

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 *hypreal-of-real* $x + u < \text{hypreal-of-real } y$
 $\langle \text{proof} \rangle$

lemma *Infinitesimal-add-hrabs-hypreal-of-real-less:*

$x \in \text{Infinitesimal} \implies |\text{hypreal-of-real } r| < \text{hypreal-of-real } y \implies$
 $|\text{hypreal-of-real } r + x| < \text{hypreal-of-real } y$
 $\langle \text{proof} \rangle$

lemma *Infinitesimal-add-hrabs-hypreal-of-real-less2:*

$x \in \text{Infinitesimal} \implies |\text{hypreal-of-real } r| < \text{hypreal-of-real } y \implies$
 $|x + \text{hypreal-of-real } r| < \text{hypreal-of-real } y$
 $\langle \text{proof} \rangle$

lemma *hypreal-of-real-le-add-Infinitesimal-cancel:*

assumes *le:* $\text{hypreal-of-real } x + u \leq \text{hypreal-of-real } y + v$
and *u:* $u \in \text{Infinitesimal}$ **and** *v:* $v \in \text{Infinitesimal}$
shows $\text{hypreal-of-real } x \leq \text{hypreal-of-real } y$
 $\langle \text{proof} \rangle$

lemma *hypreal-of-real-le-add-Infinitesimal-cancel2:*

$u \in \text{Infinitesimal} \implies v \in \text{Infinitesimal} \implies$
 $\text{hypreal-of-real } x + u \leq \text{hypreal-of-real } y + v \implies x \leq y$
 $\langle \text{proof} \rangle$

lemma *hypreal-of-real-less-Infinitesimal-le-zero:*

$\text{hypreal-of-real } x < e \implies e \in \text{Infinitesimal} \implies \text{hypreal-of-real } x \leq 0$
 $\langle \text{proof} \rangle$

lemma *Infinitesimal-add-not-zero:* $h \in \text{Infinitesimal} \implies x \neq 0 \implies \text{star-of } x + h$
 $\neq 0$

$\langle \text{proof} \rangle$

lemma *monad-hrabs-less:* $y \in \text{monad } x \implies 0 < \text{hypreal-of-real } e \implies |y - x| <$
 $\text{hypreal-of-real } e$

$\langle \text{proof} \rangle$

lemma *mem-monad-SReal-HFfinite:* $x \in \text{monad } (\text{hypreal-of-real } a) \implies x \in \text{HFfinite}$

$\langle \text{proof} \rangle$

6.7 Theorems about Standard Part

lemma *st-approx-self:* $x \in \text{HFfinite} \implies \text{st } x \approx x$

$\langle \text{proof} \rangle$

lemma *st-SReal:* $x \in \text{HFfinite} \implies \text{st } x \in \mathbb{R}$

$\langle \text{proof} \rangle$

lemma *st-HFfinite:* $x \in \text{HFfinite} \implies \text{st } x \in \text{HFfinite}$

$\langle \text{proof} \rangle$

lemma *st-unique:* $r \in \mathbb{R} \implies r \approx x \implies \text{st } x = r$

$\langle proof \rangle$

lemma *st-SReal-eq*: $x \in \mathbb{R} \implies st\ x = x$
 $\langle proof \rangle$

lemma *st-hypreal-of-real [simp]*: $st\ (hypreal\ of\ real\ x) = hypreal\ of\ real\ x$
 $\langle proof \rangle$

lemma *st-eq-approx*: $x \in HFinite \implies y \in HFinite \implies st\ x = st\ y \implies x \approx y$
 $\langle proof \rangle$

lemma *approx-st-eq*:
 assumes $x: x \in HFinite$ and $y: y \in HFinite$ and $xy: x \approx y$
 shows $st\ x = st\ y$
 $\langle proof \rangle$

lemma *st-eq-approx-iff*: $x \in HFinite \implies y \in HFinite \implies x \approx y \longleftrightarrow st\ x = st\ y$
 $\langle proof \rangle$

lemma *st-Infinitesimal-add-SReal*: $x \in \mathbb{R} \implies e \in Infinitesimal \implies st\ (x + e) = x$
 $\langle proof \rangle$

lemma *st-Infinitesimal-add-SReal2*: $x \in \mathbb{R} \implies e \in Infinitesimal \implies st\ (e + x) = x$
 $\langle proof \rangle$

lemma *HFinite-st-Infinitesimal-add*: $x \in HFinite \implies \exists e \in Infinitesimal. x = st(x) + e$
 $\langle proof \rangle$

lemma *st-add*: $x \in HFinite \implies y \in HFinite \implies st\ (x + y) = st\ x + st\ y$
 $\langle proof \rangle$

lemma *st-numeral [simp]*: $st\ (numeral\ w) = numeral\ w$
 $\langle proof \rangle$

lemma *st-neg-numeral [simp]*: $st\ (-\ numeral\ w) = -\ numeral\ w$
 $\langle proof \rangle$

lemma *st-0 [simp]*: $st\ 0 = 0$
 $\langle proof \rangle$

lemma *st-1 [simp]*: $st\ 1 = 1$
 $\langle proof \rangle$

lemma *st-neg-1 [simp]*: $st\ (-\ 1) = -\ 1$
 $\langle proof \rangle$

lemma *st-minus*: $x \in HFinite \implies st(-x) = -st x$
 $\langle proof \rangle$

lemma *st-diff*: $\llbracket x \in HFinite; y \in HFinite \rrbracket \implies st(x - y) = st x - st y$
 $\langle proof \rangle$

lemma *st-mult*: $\llbracket x \in HFinite; y \in HFinite \rrbracket \implies st(x * y) = st x * st y$
 $\langle proof \rangle$

lemma *st-Infinitesimal*: $x \in Infinitesimal \implies st x = 0$
 $\langle proof \rangle$

lemma *st-not-Infinitesimal*: $st(x) \neq 0 \implies x \notin Infinitesimal$
 $\langle proof \rangle$

lemma *st-inverse*: $x \in HFinite \implies st x \neq 0 \implies st(inverse x) = inverse(st x)$
 $\langle proof \rangle$

lemma *st-divide [simp]*: $x \in HFinite \implies y \in HFinite \implies st y \neq 0 \implies st(x / y) = st x / st y$
 $\langle proof \rangle$

lemma *st-idempotent [simp]*: $x \in HFinite \implies st(st x) = st x$
 $\langle proof \rangle$

lemma *Infinitesimal-add-st-less*:
 $x \in HFinite \implies y \in HFinite \implies u \in Infinitesimal \implies st x < st y \implies st x + u < st y$
 $\langle proof \rangle$

lemma *Infinitesimal-add-st-le-cancel*:
 $x \in HFinite \implies y \in HFinite \implies u \in Infinitesimal \implies st x \leq st y + u \implies st x \leq st y$
 $\langle proof \rangle$

lemma *st-le*: $x \in HFinite \implies y \in HFinite \implies x \leq y \implies st x \leq st y$
 $\langle proof \rangle$

lemma *st-zero-le*: $0 \leq x \implies x \in HFinite \implies 0 \leq st x$
 $\langle proof \rangle$

lemma *st-zero-ge*: $x \leq 0 \implies x \in HFinite \implies st x \leq 0$
 $\langle proof \rangle$

lemma *st-hrabs*: $x \in HFinite \implies |st x| = st |x|$
 $\langle proof \rangle$

6.8 Alternative Definitions using Free Ultrafilter

6.8.1 *HFinite*

lemma *HFinite-FreeUltrafilterNat*:

assumes $\text{star-}n\ X \in \text{HFinite}$

shows $\exists u. \text{eventually } (\lambda n. \text{norm } (X\ n) < u)\ \mathcal{U}$

$\langle \text{proof} \rangle$

lemma *FreeUltrafilterNat-HFinite*:

assumes $\text{eventually } (\lambda n. \text{norm } (X\ n) < u)\ \mathcal{U}$

shows $\text{star-}n\ X \in \text{HFinite}$

$\langle \text{proof} \rangle$

lemma *HFinite-FreeUltrafilterNat-iff*:

$\text{star-}n\ X \in \text{HFinite} \longleftrightarrow (\exists u. \text{eventually } (\lambda n. \text{norm } (X\ n) < u)\ \mathcal{U})$

$\langle \text{proof} \rangle$

6.8.2 *HInfinite*

Exclude this type of sets from free ultrafilter for Infinite numbers!

lemma *FreeUltrafilterNat-const-Finite*:

$\text{eventually } (\lambda n. \text{norm } (X\ n) = u)\ \mathcal{U} \implies \text{star-}n\ X \in \text{HFinite}$

$\langle \text{proof} \rangle$

lemma *HInfinite-FreeUltrafilterNat*:

assumes $\text{star-}n\ X \in \text{HInfinite}$ **shows** $\forall_F\ n\ \text{in } \mathcal{U}. u < \text{norm } (X\ n)$

$\langle \text{proof} \rangle$

lemma *FreeUltrafilterNat-HInfinite*:

assumes $\bigwedge u. \text{eventually } (\lambda n. u < \text{norm } (X\ n))\ \mathcal{U}$

shows $\text{star-}n\ X \in \text{HInfinite}$

$\langle \text{proof} \rangle$

lemma *HInfinite-FreeUltrafilterNat-iff*:

$\text{star-}n\ X \in \text{HInfinite} \longleftrightarrow (\forall u. \text{eventually } (\lambda n. u < \text{norm } (X\ n))\ \mathcal{U})$

$\langle \text{proof} \rangle$

6.8.3 *Infinitesimal*

lemma *ball-SReal-eq*: $(\forall x::\text{hypreal} \in \text{Reals}. P\ x) \longleftrightarrow (\forall x::\text{real}. P\ (\text{star-of } x))$

$\langle \text{proof} \rangle$

lemma *Infinitesimal-FreeUltrafilterNat-iff*:

$(\text{star-}n\ X \in \text{Infinitesimal}) = (\forall u > 0. \text{eventually } (\lambda n. \text{norm } (X\ n) < u)\ \mathcal{U})$ **(is**

$?lhs = ?rhs)$

$\langle \text{proof} \rangle$

Infinitesimals as smaller than $1/n$ for all $n::\text{nat } (> 0)$.

lemma *lemma-Infinitesimal*: $(\forall r. 0 < r \longrightarrow x < r) \longleftrightarrow (\forall n. x < \text{inverse}(\text{real}(\text{Suc } n)))$
 ⟨proof⟩

lemma *lemma-Infinitesimal2*:
 $(\forall r \in \text{Reals}. 0 < r \longrightarrow x < r) \longleftrightarrow (\forall n. x < \text{inverse}(\text{hypreal-of-nat}(\text{Suc } n)))$
 (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 ω is an infinite number

It will follow that ε 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.

ω 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 $[<X(n)] \approx \text{hypreal-of-real } a$ given that $\forall n. |X\ n - a| < 1/n$. Used in proof of $\text{NSLIM} \Rightarrow \text{LIM}$.

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*: $(\text{*f*}(\lambda n::\text{nat}. \text{inverse}(\text{real}(\text{Suc } n)))) \text{ whn} \in \text{Infinitesimal}$
 ⟨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.

$|X(n) - x| < 1/n \implies [<X\ n>] - \text{hypreal-of-real } x \in \text{Infinitesimal}$

lemma *real-seq-to-hypreal-Infinitesimal*:
 $\forall n. \text{norm } (X\ n - x) < \text{inverse}(\text{real}(\text{Suc } n)) \implies \text{star-n } X - \text{star-of } x \in \text{Infinitesimal}$

$\langle proof \rangle$

lemma *real-seq-to-hypreal-approx*:

$\forall n. \text{norm } (X\ n - x) < \text{inverse } (\text{real } (\text{Suc } n)) \implies \text{star-}n\ X \approx \text{star-of } x$
 $\langle proof \rangle$

lemma *real-seq-to-hypreal-approx2*:

$\forall n. \text{norm } (x - X\ n) < \text{inverse}(\text{real}(\text{Suc } n)) \implies \text{star-}n\ X \approx \text{star-of } x$
 $\langle proof \rangle$

lemma *real-seq-to-hypreal-Infinitesimal2*:

$\forall n. \text{norm}(X\ n - Y\ n) < \text{inverse}(\text{real}(\text{Suc } n)) \implies \text{star-}n\ X - \text{star-}n\ Y \in$
Infinitesimal
 $\langle proof \rangle$

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 *hcmmod* :: *complex star* \Rightarrow *real star*

where *hcmmod* \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 $hcomplex\text{-}of\text{-}hypreal :: hypreal \Rightarrow hcomplex$
where $hcomplex\text{-}of\text{-}hypreal \equiv of\text{-}hypreal$

definition — abbreviation for $r * (\cos a + i \sin a)$
 $hrcis :: hypreal \Rightarrow hypreal \Rightarrow hcomplex$
where $hrcis = *f2* rcis$

7.0.6 $e^{\wedge}(x + iy)$

definition $hExp :: hcomplex \Rightarrow hcomplex$
where $hExp = *f* exp$

definition $HComplex :: hypreal \Rightarrow hypreal \Rightarrow hcomplex$
where $HComplex = *f2* Complex$

lemmas $hcomplex\text{-}defs$ [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 $Standard\text{-}hRe$ [simp]: $x \in Standard \implies hRe\ x \in Standard$
 $\langle proof \rangle$

lemma $Standard\text{-}hIm$ [simp]: $x \in Standard \implies hIm\ x \in Standard$
 $\langle proof \rangle$

lemma $Standard\text{-}iii$ [simp]: $iii \in Standard$
 $\langle proof \rangle$

lemma $Standard\text{-}hcnj$ [simp]: $x \in Standard \implies hcnj\ x \in Standard$
 $\langle proof \rangle$

lemma $Standard\text{-}hsgn$ [simp]: $x \in Standard \implies hsgn\ x \in Standard$
 $\langle proof \rangle$

lemma $Standard\text{-}harg$ [simp]: $x \in Standard \implies harg\ x \in Standard$
 $\langle proof \rangle$

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

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

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

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

lemma *hcmmod-def*: $\text{hcmmod} = *f* \text{ cmod}$
 ⟨proof⟩

7.1 Properties of Nonstandard Real and Imaginary Parts

lemma *hcomplex-hRe-hIm-cancel-iff*: $\bigwedge w \ z. w = z \longleftrightarrow \text{hRe } w = \text{hRe } z \wedge \text{hIm } w = \text{hIm } z$
 ⟨proof⟩

lemma *hcomplex-equality* [intro?]: $\bigwedge z \ w. \text{hRe } z = \text{hRe } w \implies \text{hIm } z = \text{hIm } w \implies z = w$
 ⟨proof⟩

lemma *hcomplex-hRe-zero* [simp]: $\text{hRe } 0 = 0$
 ⟨proof⟩

lemma *hcomplex-hIm-zero* [simp]: $\text{hIm } 0 = 0$
 ⟨proof⟩

lemma *hcomplex-hRe-one* [simp]: $\text{hRe } 1 = 1$
 ⟨proof⟩

lemma *hcomplex-hIm-one* [simp]: $\text{hIm } 1 = 0$
 ⟨proof⟩

7.2 Addition for Nonstandard Complex Numbers

lemma *hRe-add*: $\bigwedge x \ y. \text{hRe } (x + y) = \text{hRe } x + \text{hRe } y$
 ⟨proof⟩

lemma *hIm-add*: $\bigwedge x \ y. \text{hIm } (x + y) = \text{hIm } x + \text{hIm } y$
 ⟨proof⟩

7.3 More Minus Laws

lemma *hRe-minus*: $\bigwedge z. \text{hRe } (-z) = - \text{hRe } z$
 $\langle \text{proof} \rangle$

lemma *hIm-minus*: $\bigwedge z. \text{hIm } (-z) = - \text{hIm } z$
 $\langle \text{proof} \rangle$

lemma *hcomplex-add-minus-eq-minus*: $x + y = 0 \implies x = -y$
for $x\ y :: \text{hcomplex}$
 $\langle \text{proof} \rangle$

lemma *hcomplex-i-mult-eq* [simp]: $iii * iii = -1$
 $\langle \text{proof} \rangle$

lemma *hcomplex-i-mult-left* [simp]: $\bigwedge z. iii * (iii * z) = -z$
 $\langle \text{proof} \rangle$

lemma *hcomplex-i-not-zero* [simp]: $iii \neq 0$
 $\langle \text{proof} \rangle$

7.4 More Multiplication Laws

lemma *hcomplex-mult-minus-one*: $-1 * z = -z$
for $z :: \text{hcomplex}$
 $\langle \text{proof} \rangle$

lemma *hcomplex-mult-minus-one-right*: $z * -1 = -z$
for $z :: \text{hcomplex}$
 $\langle \text{proof} \rangle$

lemma *hcomplex-mult-left-cancel*: $c \neq 0 \implies c * a = c * b \longleftrightarrow a = b$
for $a\ b\ c :: \text{hcomplex}$
 $\langle \text{proof} \rangle$

lemma *hcomplex-mult-right-cancel*: $c \neq 0 \implies a * c = b * c \longleftrightarrow a = b$
for $a\ b\ c :: \text{hcomplex}$
 $\langle \text{proof} \rangle$

7.5 Subtraction and Division

lemma *hcomplex-diff-eq-eq* [simp]: $x - y = z \longleftrightarrow x = z + y$
for $x\ y\ z :: \text{hcomplex}$
 $\langle \text{proof} \rangle$

7.6 Embedding Properties for *hcomplex-of-hypreal* Map

lemma *hRe-hcomplex-of-hypreal* [simp]: $\bigwedge z. \text{hRe } (\text{hcomplex-of-hypreal } z) = z$
 $\langle \text{proof} \rangle$

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]: $\bigwedge z. \text{HComplex} (\text{hRe } z) (\text{hIm } z) = z$
 ⟨proof⟩

lemma *hcomplex-induct* [case-names rect]:
 $(\bigwedge x y. P (\text{HComplex } x y)) \implies P z$
 ⟨proof⟩

7.8 Modulus (Absolute Value) of Nonstandard Complex Number

lemma *hcomplex-of-hypreal-abs*:
 $\text{hcomplex-of-hypreal } |x| = \text{hcomplex-of-hypreal} (\text{hcm} (\text{hcomplex-of-hypreal } x))$
 ⟨proof⟩

lemma *HComplex-inject* [simp]: $\bigwedge x y x' y'. \text{HComplex } x y = \text{HComplex } x' y' \longleftrightarrow x = x' \wedge y = y'$
 ⟨proof⟩

lemma *HComplex-add* [simp]:
 $\bigwedge x1 y1 x2 y2. \text{HComplex } x1 y1 + \text{HComplex } x2 y2 = \text{HComplex } (x1 + x2) (y1 + y2)$
 ⟨proof⟩

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

lemma *HComplex-diff* [simp]:
 $\bigwedge x1 y1 x2 y2. \text{HComplex } x1 y1 - \text{HComplex } x2 y2 = \text{HComplex } (x1 - x2) (y1 - y2)$
 ⟨proof⟩

lemma *HComplex-mult* [simp]:
 $\bigwedge x1 y1 x2 y2. \text{HComplex } x1 y1 * \text{HComplex } x2 y2 = \text{HComplex } (x1 * x2 - y1 * y2) (x1 * y2 + y1 * x2)$

$\langle \text{proof} \rangle$

HComplex-inverse is proved below.

lemma *hcomplex-of-hypreal-eq*: $\bigwedge r. \text{hcomplex-of-hypreal } r = \text{HComplex } r \ 0$
 $\langle \text{proof} \rangle$

lemma *HComplex-add-hcomplex-of-hypreal* [simp]:
 $\bigwedge x \ y \ r. \text{HComplex } x \ y + \text{hcomplex-of-hypreal } r = \text{HComplex } (x + r) \ y$
 $\langle \text{proof} \rangle$

lemma *hcomplex-of-hypreal-add-HComplex* [simp]:
 $\bigwedge r \ x \ y. \text{hcomplex-of-hypreal } r + \text{HComplex } x \ y = \text{HComplex } (r + x) \ y$
 $\langle \text{proof} \rangle$

lemma *HComplex-mult-hcomplex-of-hypreal*:
 $\bigwedge x \ y \ r. \text{HComplex } x \ y * \text{hcomplex-of-hypreal } r = \text{HComplex } (x * r) \ (y * r)$
 $\langle \text{proof} \rangle$

lemma *hcomplex-of-hypreal-mult-HComplex*:
 $\bigwedge r \ x \ y. \text{hcomplex-of-hypreal } r * \text{HComplex } x \ y = \text{HComplex } (r * x) \ (r * y)$
 $\langle \text{proof} \rangle$

lemma *i-hcomplex-of-hypreal* [simp]: $\bigwedge r. \text{iii} * \text{hcomplex-of-hypreal } r = \text{HComplex } 0 \ r$
 $\langle \text{proof} \rangle$

lemma *hcomplex-of-hypreal-i* [simp]: $\bigwedge r. \text{hcomplex-of-hypreal } r * \text{iii} = \text{HComplex } 0 \ r$
 $\langle \text{proof} \rangle$

7.9 Conjugation

lemma *hcomplex-hcnj-cancel-iff* [iff]: $\bigwedge x \ y. \text{hcnj } x = \text{hcnj } y \longleftrightarrow x = y$
 $\langle \text{proof} \rangle$

lemma *hcomplex-hcnj-hcnj* [simp]: $\bigwedge z. \text{hcnj } (\text{hcnj } z) = z$
 $\langle \text{proof} \rangle$

lemma *hcomplex-hcnj-hcomplex-of-hypreal* [simp]:
 $\bigwedge x. \text{hcnj } (\text{hcomplex-of-hypreal } x) = \text{hcomplex-of-hypreal } x$
 $\langle \text{proof} \rangle$

lemma *hcomplex-hmod-hcnj* [simp]: $\bigwedge z. \text{hcm}od (\text{hcnj } z) = \text{hcm}od z$
 $\langle \text{proof} \rangle$

lemma *hcomplex-hcnj-minus*: $\bigwedge z. \text{hcnj } (- z) = - \text{hcnj } z$
 $\langle \text{proof} \rangle$

lemma *hcomplex-hcnj-inverse*: $\bigwedge z. \text{hcnj } (\text{inverse } z) = \text{inverse } (\text{hcnj } z)$

$\langle \text{proof} \rangle$

lemma *hcomplex-hcnj-add*: $\bigwedge w z. \text{hcnj } (w + z) = \text{hcnj } w + \text{hcnj } z$
 $\langle \text{proof} \rangle$

lemma *hcomplex-hcnj-diff*: $\bigwedge w z. \text{hcnj } (w - z) = \text{hcnj } w - \text{hcnj } z$
 $\langle \text{proof} \rangle$

lemma *hcomplex-hcnj-mult*: $\bigwedge w z. \text{hcnj } (w * z) = \text{hcnj } w * \text{hcnj } z$
 $\langle \text{proof} \rangle$

lemma *hcomplex-hcnj-divide*: $\bigwedge w z. \text{hcnj } (w / z) = \text{hcnj } w / \text{hcnj } z$
 $\langle \text{proof} \rangle$

lemma *hcnj-one* [simp]: $\text{hcnj } 1 = 1$
 $\langle \text{proof} \rangle$

lemma *hcomplex-hcnj-zero* [simp]: $\text{hcnj } 0 = 0$
 $\langle \text{proof} \rangle$

lemma *hcomplex-hcnj-zero-iff* [iff]: $\bigwedge z. \text{hcnj } z = 0 \longleftrightarrow z = 0$
 $\langle \text{proof} \rangle$

lemma *hcomplex-mult-hcnj*: $\bigwedge z. z * \text{hcnj } z = \text{hcomplex-of-hypreal } ((\text{hRe } z)^2 + (\text{hIm } z)^2)$
 $\langle \text{proof} \rangle$

7.10 More Theorems about the Function *hcmmod*

lemma *hcmmod-hcomplex-of-hypreal-of-nat* [simp]:
 $\text{hcmmod } (\text{hcomplex-of-hypreal } (\text{hypreal-of-nat } n)) = \text{hypreal-of-nat } n$
 $\langle \text{proof} \rangle$

lemma *hcmmod-hcomplex-of-hypreal-of-hypnat* [simp]:
 $\text{hcmmod } (\text{hcomplex-of-hypreal}(\text{hypreal-of-hypnat } n)) = \text{hypreal-of-hypnat } n$
 $\langle \text{proof} \rangle$

lemma *hcmmod-mult-hcnj*: $\bigwedge z. \text{hcmmod } (z * \text{hcnj } z) = (\text{hcmmod } z)^2$
 $\langle \text{proof} \rangle$

lemma *hcmmod-triangle-ineq2* [simp]: $\bigwedge a b. \text{hcmmod } (b + a) - \text{hcmmod } b \leq \text{hcmmod } a$
 $\langle \text{proof} \rangle$

lemma *hcmmod-diff-ineq* [simp]: $\bigwedge a b. \text{hcmmod } a - \text{hcmmod } b \leq \text{hcmmod } (a + b)$
 $\langle \text{proof} \rangle$

7.11 Exponentiation

lemma *hcomplexpow-0* [simp]: $z \wedge 0 = 1$
 for $z :: \text{hcomplex}$

$\langle proof \rangle$

lemma *hcomplexpow-Suc* [simp]: $z \wedge (Suc\ n) = z * (z \wedge n)$
for $z :: hcomplex$
 $\langle proof \rangle$

lemma *hcomplexpow-i-squared* [simp]: $iii^2 = -1$
 $\langle proof \rangle$

lemma *hcomplex-of-hypreal-pow*: $\bigwedge x. hcomplex-of-hypreal\ (x \wedge n) = hcomplex-of-hypreal\ x \wedge n$
 $\langle proof \rangle$

lemma *hcomplex-hcnj-pow*: $\bigwedge z. hcnj\ (z \wedge n) = hcnj\ z \wedge n$
 $\langle proof \rangle$

lemma *hcmmod-hcomplexpow*: $\bigwedge x. hcmmod\ (x \wedge n) = hcmmod\ x \wedge n$
 $\langle proof \rangle$

lemma *hcpow-minus*:
 $\bigwedge x\ n. (-\ x :: hcomplex)\ pow\ n = (if\ (*p*\ even)\ n\ then\ (x\ pow\ n)\ else\ -\ (x\ pow\ n))$
 $\langle proof \rangle$

lemma *hcpow-mult*: $(r * s)\ pow\ n = (r\ pow\ n) * (s\ pow\ n)$
for $r\ s :: hcomplex$
 $\langle proof \rangle$

lemma *hcpow-zero2* [simp]: $\bigwedge n. 0\ pow\ (hSuc\ n) = (0 :: 'a :: semiring-1\ star)$
 $\langle proof \rangle$

lemma *hcpow-not-zero* [simp,intro]: $\bigwedge r\ n. r \neq 0 \implies r\ pow\ n \neq (0 :: hcomplex)$
 $\langle proof \rangle$

lemma *hcpow-zero-zero*: $r\ pow\ n = 0 \implies r = 0$
for $r :: hcomplex$
 $\langle proof \rangle$

7.12 The Function *hsgn*

lemma *hsgn-zero* [simp]: $hsgn\ 0 = 0$
 $\langle proof \rangle$

lemma *hsgn-one* [simp]: $hsgn\ 1 = 1$
 $\langle proof \rangle$

lemma *hsgn-minus*: $\bigwedge z. hsgn\ (-\ z) = -\ hsgn\ z$
 $\langle proof \rangle$

lemma *hsgn-eq*: $\bigwedge z. \text{hsgn } z = z / \text{hcomplex-of-hypreal } (\text{hcm} z)$
 $\langle \text{proof} \rangle$

lemma *hcm-i*: $\bigwedge x y. \text{hcm} (HComplex x y) = (* \text{sqrt}) (x^2 + y^2)$
 $\langle \text{proof} \rangle$

lemma *hcomplex-eq-cancel-iff1* [simp]:
 $\text{hcomplex-of-hypreal } xa = HComplex x y \longleftrightarrow xa = x \wedge y = 0$
 $\langle \text{proof} \rangle$

lemma *hcomplex-eq-cancel-iff2* [simp]:
 $HComplex x y = \text{hcomplex-of-hypreal } xa \longleftrightarrow x = xa \wedge y = 0$
 $\langle \text{proof} \rangle$

lemma *HComplex-eq-0* [simp]: $\bigwedge x y. HComplex x y = 0 \longleftrightarrow x = 0 \wedge y = 0$
 $\langle \text{proof} \rangle$

lemma *HComplex-eq-1* [simp]: $\bigwedge x y. HComplex x y = 1 \longleftrightarrow x = 1 \wedge y = 0$
 $\langle \text{proof} \rangle$

lemma *i-eq-HComplex-0-1*: $iii = HComplex 0 1$
 $\langle \text{proof} \rangle$

lemma *HComplex-eq-i* [simp]: $\bigwedge x y. HComplex x y = iii \longleftrightarrow x = 0 \wedge y = 1$
 $\langle \text{proof} \rangle$

lemma *hRe-hsgn* [simp]: $\bigwedge z. \text{hRe } (\text{hsgn } z) = \text{hRe } z / \text{hcm} z$
 $\langle \text{proof} \rangle$

lemma *hIm-hsgn* [simp]: $\bigwedge z. \text{hIm } (\text{hsgn } z) = \text{hIm } z / \text{hcm} z$
 $\langle \text{proof} \rangle$

lemma *HComplex-inverse*: $\bigwedge x y. \text{inverse } (HComplex x y) = HComplex (x / (x^2 + y^2)) (-y / (x^2 + y^2))$
 $\langle \text{proof} \rangle$

lemma *hRe-mult-i-eq*[simp]: $\bigwedge y. \text{hRe } (iii * \text{hcomplex-of-hypreal } y) = 0$
 $\langle \text{proof} \rangle$

lemma *hIm-mult-i-eq* [simp]: $\bigwedge y. \text{hIm } (iii * \text{hcomplex-of-hypreal } y) = y$
 $\langle \text{proof} \rangle$

lemma *hcm-mult-i* [simp]: $\bigwedge y. \text{hcm} (iii * \text{hcomplex-of-hypreal } y) = |y|$
 $\langle \text{proof} \rangle$

lemma *hcm-mult-i2* [simp]: $\bigwedge y. \text{hcm} (\text{hcomplex-of-hypreal } y * iii) = |y|$
 $\langle \text{proof} \rangle$

7.12.1 *harg*

lemma *cos-harg-i-mult-zero* [simp]: $\bigwedge y. y \neq 0 \implies (*f* \cos) (harg (HComplex 0 y)) = 0$
 ⟨proof⟩

7.13 Polar Form for Nonstandard Complex Numbers

lemma *complex-split-polar2*: $\forall n. \exists r a. (z\ n) = \text{complex-of-real } r * \text{Complex } (\cos a) (\sin a)$
 ⟨proof⟩

lemma *hcomplex-split-polar*:

$\bigwedge z. \exists r a. z = \text{hcomplex-of-hypreal } r * (HComplex ((*f* \cos) a) ((*f* \sin) a))$
 ⟨proof⟩

lemma *hcis-eq*:

$\bigwedge a. \text{hcis } a = \text{hcomplex-of-hypreal } ((*f* \cos) a) + iii * \text{hcomplex-of-hypreal } ((*f* \sin) a)$
 ⟨proof⟩

lemma *hrcis-Ex*: $\bigwedge z. \exists r a. z = \text{hrcis } r\ a$
 ⟨proof⟩

lemma *hRe-hcomplex-polar* [simp]:

$\bigwedge r a. \text{hRe } (\text{hcomplex-of-hypreal } r * HComplex ((*f* \cos) a) ((*f* \sin) a)) = r * (*f* \cos) a$
 ⟨proof⟩

lemma *hRe-hrcis* [simp]: $\bigwedge r a. \text{hRe } (\text{hrcis } r\ a) = r * (*f* \cos) a$
 ⟨proof⟩

lemma *hIm-hcomplex-polar* [simp]:

$\bigwedge r a. \text{hIm } (\text{hcomplex-of-hypreal } r * HComplex ((*f* \cos) a) ((*f* \sin) a)) = r * (*f* \sin) a$
 ⟨proof⟩

lemma *hIm-hrcis* [simp]: $\bigwedge r a. \text{hIm } (\text{hrcis } r\ a) = r * (*f* \sin) a$
 ⟨proof⟩

lemma *hcmmod-unit-one* [simp]: $\bigwedge a. \text{hcmmod } (HComplex ((*f* \cos) a) ((*f* \sin) a)) = 1$
 ⟨proof⟩

lemma *hcmmod-complex-polar* [simp]:

$\bigwedge r a. \text{hcmmod } (\text{hcomplex-of-hypreal } r * HComplex ((*f* \cos) a) ((*f* \sin) a)) = |r|$
 ⟨proof⟩

lemma *hcmmod-hrcis* [simp]: $\bigwedge r a. \text{hcmmod}(\text{hrcis } r\ a) = |r|$

$\langle proof \rangle$

$$(r1 * hrcis\ a) * (r2 * hrcis\ b) = r1 * r2 * hrcis\ (a + b)$$

lemma *hcis-hrcis-eq*: $\bigwedge a. hcis\ a = hrcis\ 1\ a$

$\langle proof \rangle$

declare *hcis-hrcis-eq* [*symmetric, simp*]

lemma *hrcis-mult*: $\bigwedge a\ b\ r1\ r2. hrcis\ r1\ a * hrcis\ r2\ b = hrcis\ (r1 * r2)\ (a + b)$

$\langle proof \rangle$

lemma *hcis-mult*: $\bigwedge a\ b. hcis\ a * hcis\ b = hcis\ (a + b)$

$\langle proof \rangle$

lemma *hcis-zero* [*simp*]: $hcis\ 0 = 1$

$\langle proof \rangle$

lemma *hrcis-zero-mod* [*simp*]: $\bigwedge a. hrcis\ 0\ a = 0$

$\langle proof \rangle$

lemma *hrcis-zero-arg* [*simp*]: $\bigwedge r. hrcis\ r\ 0 = hcomplex-of-hypreal\ r$

$\langle proof \rangle$

lemma *hcomplex-i-mult-minus* [*simp*]: $\bigwedge x. iii * (iii * x) = -\ x$

$\langle proof \rangle$

lemma *hcomplex-i-mult-minus2* [*simp*]: $iii * iii * x = -\ x$

$\langle proof \rangle$

lemma *hcis-hypreal-of-nat-Suc-mult*:

$$\bigwedge a. hcis\ (hypreal-of-nat\ (Suc\ n) * a) = hcis\ a * hcis\ (hypreal-of-nat\ n * a)$$

$\langle proof \rangle$

lemma *NSDeMoivre*: $\bigwedge a. (hcis\ a) \wedge^n = hcis\ (hypreal-of-nat\ n * a)$

$\langle proof \rangle$

lemma *hcis-hypreal-of-hypnat-Suc-mult*:

$$\bigwedge a\ n. hcis\ (hypreal-of-hypnat\ (n + 1) * a) = hcis\ a * hcis\ (hypreal-of-hypnat\ n * a)$$

$\langle proof \rangle$

lemma *NSDeMoivre-ext*: $\bigwedge a\ n. (hcis\ a) \text{ pow } n = hcis\ (hypreal-of-hypnat\ n * a)$

$\langle proof \rangle$

lemma *NSDeMoivre2*: $\bigwedge a\ r. (hrcis\ r\ a) \wedge^n = hrcis\ (r \wedge n)\ (hypreal-of-nat\ n * a)$

$\langle proof \rangle$

lemma *DeMoivre2-ext*: $\bigwedge a\ r\ n. (hrcis\ r\ a) \text{ pow } n = hrcis\ (r \text{ pow } n)\ (hypreal-of-hypnat\ n * a)$

$\langle proof \rangle$

lemma *hcis-inverse [simp]*: $\bigwedge a. \text{inverse } (\text{hcis } a) = \text{hcis } (- a)$
 $\langle \text{proof} \rangle$

lemma *hrcis-inverse*: $\bigwedge a \ r. \text{inverse } (\text{hrcis } r \ a) = \text{hrcis } (\text{inverse } r) \ (- a)$
 $\langle \text{proof} \rangle$

lemma *hRe-hcis [simp]*: $\bigwedge a. \text{hRe } (\text{hcis } a) = (*f* \cos) \ a$
 $\langle \text{proof} \rangle$

lemma *hIm-hcis [simp]*: $\bigwedge a. \text{hIm } (\text{hcis } a) = (*f* \sin) \ a$
 $\langle \text{proof} \rangle$

lemma *cos-n-hRe-hcis-pow-n*: $(*f* \cos) (\text{hypreal-of-nat } n * a) = \text{hRe } (\text{hcis } a ^ n)$
 $\langle \text{proof} \rangle$

lemma *sin-n-hIm-hcis-pow-n*: $(*f* \sin) (\text{hypreal-of-nat } n * a) = \text{hIm } (\text{hcis } a ^ n)$
 $\langle \text{proof} \rangle$

lemma *cos-n-hRe-hcis-hcpow-n*: $(*f* \cos) (\text{hypreal-of-hypnat } n * a) = \text{hRe } (\text{hcis } a \text{ pow } n)$
 $\langle \text{proof} \rangle$

lemma *sin-n-hIm-hcis-hcpow-n*: $(*f* \sin) (\text{hypreal-of-hypnat } n * a) = \text{hIm } (\text{hcis } a \text{ pow } n)$
 $\langle \text{proof} \rangle$

lemma *hExp-add*: $\bigwedge a \ b. \text{hExp } (a + b) = \text{hExp } a * \text{hExp } b$
 $\langle \text{proof} \rangle$

7.14 *hcomplex-of-complex*: the Injection from type *complex* to *hcomplex*

lemma *hcomplex-of-complex-i*: $\text{iii} = \text{hcomplex-of-complex } \text{i}$
 $\langle \text{proof} \rangle$

lemma *hRe-hcomplex-of-complex*: $\text{hRe } (\text{hcomplex-of-complex } z) = \text{hypreal-of-real } (\text{Re } z)$
 $\langle \text{proof} \rangle$

lemma *hIm-hcomplex-of-complex*: $\text{hIm } (\text{hcomplex-of-complex } z) = \text{hypreal-of-real } (\text{Im } z)$
 $\langle \text{proof} \rangle$

lemma *hcmmod-hcomplex-of-complex*: $\text{hcmmod } (\text{hcomplex-of-complex } x) = \text{hypreal-of-real } (\text{cmmod } x)$
 $\langle \text{proof} \rangle$

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 (− numeral w) = hcomplex-of-hypreal (− numeral w)
<proof>

lemma *hcomplex-numeral-hcnj [simp]*: *hcnj (numeral v :: 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 ⇒ 'a set) ⇒ 'a star set
*(⟨⟨open-block notation=⟨prefix starset-n⟩*sn* -⟩ [80] 80)*
where **sn* As = Iset (star-n As)*

definition *InternalSets :: 'a star set set*

where *InternalSets = {X. ∃ As. X = *sn* As}*

definition — nonstandard extension of function

is-starext :: ('a star ⇒ 'a star) ⇒ ('a ⇒ 'a) ⇒ 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

$starfun-n :: (nat \Rightarrow 'a \Rightarrow 'b) \Rightarrow 'a\ star \Rightarrow 'b\ star$
 $(\langle \langle open-block\ notation = \langle prefix\ starfun-n \rangle \rangle *fn* - \rangle [80]\ 80)$
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 $no-choice: \forall x. \exists y. Q\ x\ y \implies \exists f :: 'a \Rightarrow nat. \forall x. Q\ x\ (f\ x)$
 $\langle proof \rangle$

8.2 Properties of the Star-transform Applied to Sets of Reals

lemma $STAR-star-of-image-subset: star-of\ 'A \subseteq *s* A$
 $\langle proof \rangle$

lemma $STAR-hypreal-of-real-Int: *s* X \cap \mathbb{R} = hypreal-of-real\ 'X$
 $\langle proof \rangle$

lemma $STAR-star-of-Int: *s* X \cap Standard = star-of\ 'X$
 $\langle proof \rangle$

lemma $lemma-not-hyprealA: x \notin hypreal-of-real\ 'A \implies \forall y \in A. x \neq hypreal-of-real\ y$
 $\langle proof \rangle$

lemma $lemma-not-starA: x \notin star-of\ 'A \implies \forall y \in A. x \neq star-of\ y$
 $\langle proof \rangle$

lemma $STAR-real-seq-to-hypreal: \forall n. (X\ n) \notin M \implies star-n\ X \notin *s* M$
 $\langle proof \rangle$

lemma $STAR-singleton: *s* \{x\} = \{star-of\ x\}$
 $\langle proof \rangle$

lemma $STAR-not-mem: x \notin F \implies star-of\ x \notin *s* F$
 $\langle proof \rangle$

lemma $STAR-subset-closed: x \in *s* A \implies A \subseteq B \implies x \in *s* B$
 $\langle proof \rangle$

Nonstandard extension of a set (defined using a constant sequence) as a special case of an internal set.

lemma *starset-n-starset*: $\forall n. As\ n = A \implies *sn* As = *s* A$
 $\langle proof \rangle$

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*: $F = (\lambda n. f) \implies *fn* F = *f* f$
 $\langle proof \rangle$

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*
 $\langle proof \rangle$

Nonstandard extension of functions.

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

lemma *starfun-if-eq*: $\bigwedge w. w \neq star\text{-}of\ x \implies (*f* (\lambda z. if\ z = x\ then\ a\ else\ g\ z))$
 $w = (*f* g)\ w$
 $\langle proof \rangle$

Multiplication: $(*f)\ x\ (*g) = *(f\ x\ g)$

lemma *starfun-mult*: $\bigwedge x. (*f* f)\ x * (*f* g)\ x = (*f* (\lambda x. f\ x * g\ x))\ x$
 $\langle proof \rangle$

declare *starfun-mult* [*symmetric, simp*]

Addition: $(*f) + (*g) = *(f + g)$

lemma *starfun-add*: $\bigwedge x. (*f* f)\ x + (*f* g)\ x = (*f* (\lambda x. f\ x + g\ x))\ x$
 $\langle proof \rangle$

declare *starfun-add* [*symmetric, simp*]

Subtraction: $(*f) + -(*g) = *(f + -g)$

lemma *starfun-minus*: $\bigwedge x. - (*f* f)\ x = (*f* (\lambda x. - f\ x))\ x$
 $\langle proof \rangle$

declare *starfun-minus* [*symmetric, simp*]

lemma *starfun-add-minus*: $\bigwedge x. (*f* f)\ x + -(*f* g)\ x = (*f* (\lambda x. f\ x + -g\ x))\ x$
 $\langle proof \rangle$

declare *starfun-add-minus* [*symmetric*, *simp*]

lemma *starfun-diff*: $\bigwedge x. (*f* f) x - (*f* g) x = (*f* (\lambda x. f x - g x)) x$
 $\langle proof \rangle$

declare *starfun-diff* [*symmetric*, *simp*]

Composition: $(*f) \circ (*g) = *(f \circ g)$

lemma *starfun-o2*: $(\lambda x. (*f* f) ((*f* g) x)) = *f* (\lambda x. f (g x))$
 $\langle proof \rangle$

lemma *starfun-o*: $(*f* f) \circ (*f* g) = (*f* (f \circ g))$
 $\langle proof \rangle$

NS extension of constant function.

lemma *starfun-const-fun* [*simp*]: $\bigwedge x. (*f* (\lambda x. k)) x = \text{star-of } k$
 $\langle proof \rangle$

The NS extension of the identity function.

lemma *starfun-Id* [*simp*]: $\bigwedge x. (*f* (\lambda x. x)) x = x$
 $\langle proof \rangle$

The Star-function is a (nonstandard) extension of the function.

lemma *is-starext-starfun*: *is-starext* $(*f* f) f$
 $\langle proof \rangle$

Any nonstandard extension is in fact the Star-function.

lemma *is-starfun-starext*:

assumes *is-starext* $F f$

shows $F = *f* f$

$\langle proof \rangle$

lemma *is-starext-starfun-iff*: *is-starext* $F f \longleftrightarrow F = *f* f$
 $\langle proof \rangle$

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*: $(*f* f) (\text{star-of } a) = \text{star-of } (f a)$
 $\langle proof \rangle$

lemma *starfun-approx*: $(*f* f) (\text{star-of } a) \approx \text{star-of } (f a)$
 $\langle proof \rangle$

Useful for NS definition of derivatives.

lemma *starfun-lambda-cancel*: $\bigwedge x'. (*f* (\lambda h. f (x + h))) x' = (*f* f) (\text{star-of } x + x')$
 $\langle proof \rangle$

lemma *starfun-lambda-cancel2*: $(\text{*f* } (\lambda h. f (g (x + h)))) x' = (\text{*f* } (f \circ g))$
(star-of $x + x')$
<proof>

lemma *starfun-mult-HFinite-approx*:
 $(\text{*f* } f) x \approx l \implies (\text{*f* } g) x \approx m \implies l \in HFinite \implies m \in HFinite \implies$
 $(\text{*f* } (\lambda x. f x * g x)) x \approx l * m$
for $l m :: 'a::\text{real-normed-algebra star}$
<proof>

lemma *starfun-add-approx*: $(\text{*f* } f) x \approx l \implies (\text{*f* } g) x \approx m \implies (\text{*f* } (\%x. f x + g x)) x \approx l + m$
<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*: $\text{*f* } abs = abs$
<proof>

lemma *starfun-inverse-inverse* [*simp*]: $(\text{*f* } inverse) x = inverse x$
<proof>

lemma *starfun-inverse*: $\bigwedge x. inverse ((\text{*f* } f) x) = (\text{*f* } (\lambda x. inverse (f x))) x$
<proof>

declare *starfun-inverse* [*symmetric, simp*]

lemma *starfun-divide*: $\bigwedge x. (\text{*f* } f) x / (\text{*f* } g) x = (\text{*f* } (\lambda x. f x / g x)) x$
<proof>

declare *starfun-divide* [*symmetric, simp*]

lemma *starfun-inverse2*: $\bigwedge x. inverse ((\text{*f* } f) x) = (\text{*f* } (\lambda x. inverse (f x))) x$
<proof>

General lemma/theorem needed for proofs in elementary topology of the reals.

lemma *starfun-mem-starset*: $\bigwedge x. (\text{*f* } f) x \in ** A \implies x \in ** \{x. f x \in A\}$
<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 extenson. See second NS set extension below.

lemma *STAR-rabs-add-minus*: $*s* \{x. |x + - y| < r\} = \{x. |x + -star-of y| < star-of r\}$
 $\langle proof \rangle$

lemma *STAR-starfun-rabs-add-minus*:
 $*s* \{x. |f x + - y| < r\} = \{x. |(*f* f) x + -star-of y| < star-of r\}$
 $\langle proof \rangle$

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*:
 $star-n X \in Infinitesimal \longleftrightarrow (\forall m. eventually (\lambda n. norm (X n) < inverse (real (Suc m)))) \mathcal{U}$
 $\langle proof \rangle$

lemma *HNatInfinite-inverse-Infinitesimal [simp]*:
assumes $n \in HNatInfinite$
shows $inverse (hypreal-of-hypnat n) \in Infinitesimal$
 $\langle proof \rangle$

lemma *approx-FreeUltrafilterNat-iff*:
 $star-n X \approx star-n Y \longleftrightarrow (\forall r > 0. eventually (\lambda n. norm (X n - Y n) < r) \mathcal{U})$
(is ?lhs = ?rhs)
 $\langle proof \rangle$

lemma *approx-FreeUltrafilterNat-iff2*:
 $star-n X \approx star-n Y \longleftrightarrow (\forall m. eventually (\lambda n. norm (X n - Y n) < inverse (real (Suc m)))) \mathcal{U}$
(is ?lhs = ?rhs)
 $\langle proof \rangle$

lemma *inj-starfun*: $inj starfun$
 $\langle proof \rangle$

end

9 Star-transforms for the Hypernaturals

theory *NatStar*
imports *Star*
begin

lemma *star-n-eq-starfun-whn*: $star-n X = (*f* X) \text{ whn}$
 $\langle proof \rangle$

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*: $*sn* (\lambda n. A \cap B \ n) = *sn* A \cap *sn* B$
 $\langle \text{proof} \rangle$

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*: $*sn* ((\lambda n. \neg A \ n)) = \neg (*sn* A)$
 $\langle \text{proof} \rangle$

lemma *InternalSets-Compl*: $X \in \text{InternalSets} \implies \neg X \in \text{InternalSets}$
 $\langle \text{proof} \rangle$

lemma *starset-n-diff*: $*sn* (\lambda n. (A \ n) - (B \ n)) = *sn* A - *sn* B$
 $\langle \text{proof} \rangle$

lemma *InternalSets-diff*: $X \in \text{InternalSets} \implies Y \in \text{InternalSets} \implies X - Y \in \text{InternalSets}$
 $\langle \text{proof} \rangle$

lemma *NatStar-SHNat-subset*: $\text{Nats} \leq *s* (\text{UNIV}:: \text{nat set})$
 $\langle \text{proof} \rangle$

lemma *NatStar-hypreal-of-real-Int*: $*s* X \ \text{Int} \ \text{Nats} = \text{hypnat-of-nat} \ ` X$
 $\langle \text{proof} \rangle$

lemma *starset-starset-n-eq*: $*s* X = *sn* (\lambda n. X)$
 $\langle \text{proof} \rangle$

lemma *InternalSets-starset-n [simp]*: $(*s* X) \in \text{InternalSets}$
 $\langle \text{proof} \rangle$

lemma *InternalSets-UNIV-diff*: $X \in \text{InternalSets} \implies \text{UNIV} - X \in \text{InternalSets}$
 $\langle \text{proof} \rangle$

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*: $\forall n. N \leq n \longrightarrow f \ n \leq g \ n \implies \forall n. \text{hypnat-of-nat} \ N \leq n \longrightarrow (*f* f) \ n \leq (*f* g) \ n$
 $\langle \text{proof} \rangle$

And another:

lemma *starfun-less-mono*:

$\forall n. N \leq n \longrightarrow f\ n < g\ n \implies \forall n. \text{hypnat-of-nat } N \leq n \longrightarrow (*f* f)\ n < (*f* g)\ n$
 $\langle \text{proof} \rangle$

Nonstandard extension when we increment the argument by one.

lemma *starfun-shift-one*: $\bigwedge N. (*f* (\lambda n. f\ (Suc\ n)))\ N = (*f* f)\ (N + (1::\text{hypnat}))$
 $\langle \text{proof} \rangle$

Nonstandard extension with absolute value.

lemma *starfun-abs*: $\bigwedge N. (*f* (\lambda n. |f\ n|))\ N = |(*f* f)\ N|$
 $\langle \text{proof} \rangle$

The *hyperpow* function as a nonstandard extension of *realpow*.

lemma *starfun-pow*: $\bigwedge N. (*f* (\lambda n. r\ ^n))\ N = \text{hypreal-of-real } r\ \text{pow } N$
 $\langle \text{proof} \rangle$

lemma *starfun-pow2*: $\bigwedge N. (*f* (\lambda n. X\ n\ ^m))\ N = (*f* X)\ N\ \text{pow } \text{hypnat-of-nat } m$
 $\langle \text{proof} \rangle$

lemma *starfun-pow3*: $\bigwedge R. (*f* (\lambda r. r\ ^n))\ R = R\ \text{pow } \text{hypnat-of-nat } n$
 $\langle \text{proof} \rangle$

The *hypreal-of-hypnat* function as a nonstandard extension of *real*.

lemma *starfunNat-real-of-nat*: $(*f* \text{real}) = \text{hypreal-of-hypnat}$
 $\langle \text{proof} \rangle$

lemma *starfun-inverse-real-of-nat-eq*:

$N \in \text{HNatInfinite} \implies (*f* (\lambda x::\text{nat}. \text{inverse } (\text{real } x)))\ N = \text{inverse } (\text{hypreal-of-hypnat } N)$
 $\langle \text{proof} \rangle$

Internal functions – some redundancy with **f** now.

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

Multiplication: $(*fn)\ x\ (*gn) = *(fn\ x\ gn)$

lemma *starfun-n-mult*: $(*fn* f)\ z\ (*fn* g)\ z = (*fn* (\lambda i\ x. f\ i\ x\ *g\ i\ x))\ z$
 $\langle \text{proof} \rangle$

Addition: $(*fn) + (*gn) = *(fn + gn)$

lemma *starfun-n-add*: $(*fn* f)\ z + (*fn* g)\ z = (*fn* (\lambda i\ x. f\ i\ x + g\ i\ x))\ z$
 $\langle \text{proof} \rangle$

Subtraction: $(*fn) - (*gn) = *(fn + -\ gn)$

lemma *starfun-n-add-minus*: $(\text{*fn* } f) \ z + -(\text{*fn* } g) \ z = (\text{*fn* } (\lambda i \ x. f \ i \ x + -g \ i \ x)) \ z$
 ⟨proof⟩

Composition: $(\text{*fn* } f) \circ (\text{*gn* } g) = \text{*}(fn \circ gn)$

lemma *starfun-n-const-fun* [simp]: $(\text{*fn* } (\lambda i \ x. k)) \ z = \text{star-of } k$
 ⟨proof⟩

lemma *starfun-n-minus*: $-(\text{*fn* } f) \ x = (\text{*fn* } (\lambda i \ x. -(f \ i) \ x)) \ x$
 ⟨proof⟩

lemma *starfun-n-eq* [simp]: $(\text{*fn* } f) \ (\text{star-of } n) = \text{star-n } (\lambda i. f \ i \ n)$
 ⟨proof⟩

lemma *starfun-eq-iff*: $((\text{*f* } f) = (\text{*f* } g)) \longleftrightarrow f = g$
 ⟨proof⟩

lemma *starfunNat-inverse-real-of-nat-Infinitesimal* [simp]:
 $N \in HNatInfinite \implies (\text{*f* } (\lambda x. \text{inverse } (\text{real } x))) \ N \in Infinitesimal$
 ⟨proof⟩

9.2 Nonstandard Characterization of Induction

lemma *hypnat-induct-obj*:
 $\bigwedge n. ((\text{*p* } P) \ (0::\text{hypnat}) \wedge (\forall n. (\text{*p* } P) \ n \longrightarrow (\text{*p* } P) \ (n + 1))) \longrightarrow (\text{*p* } P) \ n$
 ⟨proof⟩

lemma *hypnat-induct*:
 $\bigwedge n. (\text{*p* } P) \ (0::\text{hypnat}) \implies (\bigwedge n. (\text{*p* } P) \ n \implies (\text{*p* } P) \ (n + 1)) \implies (\text{*p* } P) \ n$
 ⟨proof⟩

lemma *starP2-eq-iff*: $(\text{*p2* } (=)) = (=)$
 ⟨proof⟩

lemma *starP2-eq-iff2*: $(\text{*p2* } (\lambda x \ y. x = y)) \ X \ Y \longleftrightarrow X = Y$
 ⟨proof⟩

lemma *nonempty-set-star-has-least-lemma*:
 $\exists n \in S. \forall m \in S. n \leq m \text{ if } S \neq \{\} \text{ for } S :: \text{nat set}$
 ⟨proof⟩

lemma *nonempty-set-star-has-least*:
 $\bigwedge S::\text{nat set star}. \text{Iset } S \neq \{\} \implies \exists n \in \text{Iset } S. \forall m \in \text{Iset } S. n \leq m$
 ⟨proof⟩

lemma *nonempty-InternalNatSet-has-least*: $S \in \text{InternalSets} \implies S \neq \{\} \implies \exists n \in S. \forall m \in S. n \leq m$

for $S :: \text{hypnat set}$
 $\langle \text{proof} \rangle$

Goldblatt, page 129 Thm 11.3.2.

lemma *internal-induct-lemma*:

$\bigwedge X :: \text{nat set star}.$
 $(0 :: \text{hypnat}) \in \text{Iset } X \implies \forall n. n \in \text{Iset } X \longrightarrow n + 1 \in \text{Iset } X \implies \text{Iset } X =$
 $(\text{UNIV} :: \text{hypnat set})$
 $\langle \text{proof} \rangle$

lemma *internal-induct*:

$X \in \text{InternalSets} \implies (0 :: \text{hypnat}) \in X \implies \forall n. n \in X \longrightarrow n + 1 \in X \implies X =$
 $(\text{UNIV} :: \text{hypnat set})$
 $\langle \text{proof} \rangle$

end

10 Sequences and Convergence (Nonstandard)

theory *HSEQ*

imports *Complex-Main NatStar*

abbrevs $----> = \longrightarrow_{NS}$

begin

definition *NSLIMSEQ* :: $(\text{nat} \Rightarrow 'a :: \text{real-normed-vector}) \Rightarrow 'a \Rightarrow \text{bool}$
 $(\langle \langle \text{notation} = \langle \text{mixfix } \text{NSLIMSEQ} \rangle \rangle (-) / \longrightarrow_{NS} (-) \rangle [60, 60] 60)$ **where**
 — Nonstandard definition of convergence of sequence
 $X \longrightarrow_{NS} L \longleftrightarrow (\forall N \in \text{HNatInfinite}. (*f* X) N \approx \text{star-of } L)$

definition *nslim* :: $(\text{nat} \Rightarrow 'a :: \text{real-normed-vector}) \Rightarrow 'a$
where $\text{nslim } X = (\text{THE } L. X \longrightarrow_{NS} L)$
 — Nonstandard definition of limit using choice operator

definition *NSconvergent* :: $(\text{nat} \Rightarrow 'a :: \text{real-normed-vector}) \Rightarrow \text{bool}$
where $\text{NSconvergent } X \longleftrightarrow (\exists L. X \longrightarrow_{NS} L)$
 — Nonstandard definition of convergence

definition *NSBseq* :: $(\text{nat} \Rightarrow 'a :: \text{real-normed-vector}) \Rightarrow \text{bool}$
where $\text{NSBseq } X \longleftrightarrow (\forall N \in \text{HNatInfinite}. (*f* X) N \in \text{HFinite})$
 — Nonstandard definition for bounded sequence

definition *NSCauchy* :: $(\text{nat} \Rightarrow 'a :: \text{real-normed-vector}) \Rightarrow \text{bool}$
where $\text{NSCauchy } X \longleftrightarrow (\forall M \in \text{HNatInfinite}. \forall N \in \text{HNatInfinite}. (*f* X) M \approx (*f* X) N)$
 — Nonstandard definition

10.1 Limits of Sequences

lemma *NSLIMSEQ-I*: $(\bigwedge N. N \in \text{HNatInfinite} \implies \text{starfun } X \ N \approx \text{star-of } L) \implies$
 $X \longrightarrow_{NS} L$
 ⟨proof⟩

lemma *NSLIMSEQ-D*: $X \longrightarrow_{NS} L \implies N \in \text{HNatInfinite} \implies \text{starfun } X \ N \approx$
 $\text{star-of } L$
 ⟨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 $a \ b :: 'a::\text{real-normed-algebra}$
 ⟨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*: $X \longrightarrow_{NS} a \implies a \neq 0 \implies (\lambda n. \text{inverse } (X \ n))$
 $\longrightarrow_{NS} \text{inverse } a$
for $a :: 'a::\text{real-normed-div-algebra}$
 ⟨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 $a \ b :: 'a::\text{real-normed-field}$
 ⟨proof⟩

lemma *starfun-hnorm*: $\bigwedge x. \text{hnorm } ((* f * f) \ x) = (* f * (\lambda x. \text{norm } (f \ x))) \ x$
 ⟨proof⟩

lemma *NSLIMSEQ-norm*: $X \longrightarrow_{NS} a \implies (\lambda n. \text{norm } (X\ n)) \longrightarrow_{NS} \text{norm } a$
 $\langle \text{proof} \rangle$

Uniqueness of limit.

lemma *NSLIMSEQ-unique*: $X \longrightarrow_{NS} a \implies X \longrightarrow_{NS} b \implies a = b$
 $\langle \text{proof} \rangle$

lemma *NSLIMSEQ-pow* [rule-format]: $(X \longrightarrow_{NS} a) \longrightarrow ((\lambda n. (X\ n) \wedge m) \longrightarrow_{NS} a \wedge m)$
for $a :: 'a :: \{\text{real-normed-algebra}, \text{power}\}$
 $\langle \text{proof} \rangle$

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 $l\ m :: \text{real}$
 $\langle \text{proof} \rangle$

lemma *NSLIMSEQ-le-const*: $X \longrightarrow_{NS} r \implies \forall n. a \leq X\ n \implies a \leq r$
for $a\ r :: \text{real}$
 $\langle \text{proof} \rangle$

lemma *NSLIMSEQ-le-const2*: $X \longrightarrow_{NS} r \implies \forall n. X\ n \leq a \implies r \leq a$
for $a\ r :: \text{real}$
 $\langle \text{proof} \rangle$

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*: $((\lambda n. f\ (\text{Suc } n)) \longrightarrow_{NS} l) \longleftrightarrow (f \longrightarrow_{NS} l)$
 $\langle \text{proof} \rangle$

10.1.1 Equivalence of LIMSEQ and NSLIMSEQ

lemma *LIMSEQ-NSLIMSEQ*:
assumes $X: X \longrightarrow L$
shows $X \longrightarrow_{NS} L$
 $\langle \text{proof} \rangle$

lemma *NSLIMSEQ-LIMSEQ*:
assumes $X: X \longrightarrow_{NS} L$
shows $X \longrightarrow L$
 $\langle \text{proof} \rangle$

theorem *LIMSEQ-NSLIMSEQ-iff*: $f \longrightarrow L \longleftrightarrow f \longrightarrow_{NS} L$
 $\langle \text{proof} \rangle$

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*: $(\lambda n. \text{norm } (X\ n)) \longrightarrow_{NS} 0 \longleftrightarrow X \longrightarrow_{NS} 0$
 $\langle \text{proof} \rangle$

lemma *NSLIMSEQ-rabs-zero*: $(\lambda n. |f\ n|) \longrightarrow_{NS} 0 \longleftrightarrow f \longrightarrow_{NS} (0::\text{real})$
 $\langle \text{proof} \rangle$

Generalization to other limits.

lemma *NSLIMSEQ-imp-rabs*: $f \longrightarrow_{NS} l \implies (\lambda n. |f\ n|) \longrightarrow_{NS} |l|$
for $l :: \text{real}$
 $\langle \text{proof} \rangle$

lemma *NSLIMSEQ-inverse-zero*: $\forall y::\text{real}. \exists N. \forall n \geq N. y < f\ n \implies (\lambda n. \text{inverse } (f\ n)) \longrightarrow_{NS} 0$
 $\langle \text{proof} \rangle$

lemma *NSLIMSEQ-inverse-real-of-nat*: $(\lambda n. \text{inverse } (\text{real } (\text{Suc } n))) \longrightarrow_{NS} 0$
 $\langle \text{proof} \rangle$

lemma *NSLIMSEQ-inverse-real-of-nat-add*: $(\lambda n. r + \text{inverse } (\text{real } (\text{Suc } n))) \longrightarrow_{NS} r$
 $\langle \text{proof} \rangle$

lemma *NSLIMSEQ-inverse-real-of-nat-add-minus*: $(\lambda n. r + - \text{inverse } (\text{real } (\text{Suc } n))) \longrightarrow_{NS} r$
 $\langle \text{proof} \rangle$

lemma *NSLIMSEQ-inverse-real-of-nat-add-minus-mult*:
 $(\lambda n. r * (1 + - \text{inverse } (\text{real } (\text{Suc } n)))) \longrightarrow_{NS} r$
 $\langle \text{proof} \rangle$

10.2 Convergence

lemma *nslimI*: $X \longrightarrow_{NS} L \implies \text{nslim } X = L$
 $\langle \text{proof} \rangle$

lemma *lim-nslim-iff*: $\text{lim } X = \text{nslim } X$
 $\langle \text{proof} \rangle$

lemma *NSconvergentD*: $\text{NSconvergent } X \implies \exists L. X \longrightarrow_{NS} L$
 $\langle \text{proof} \rangle$

lemma *NSconvergentI*: $X \longrightarrow_{NS} L \implies \text{NSconvergent } X$
 $\langle \text{proof} \rangle$

lemma *convergent-NSconvergent-iff*: $\text{convergent } X = \text{NSconvergent } X$
 $\langle \text{proof} \rangle$

lemma *NSconvergent-NSLIMSEQ-iff*: $\text{NSconvergent } X \longleftrightarrow X \longrightarrow_{NS} \text{nslim } X$
 $\langle \text{proof} \rangle$

10.3 Bounded Monotonic Sequences

lemma *NSBseqD*: $\text{NSBseq } X \implies N \in \text{HNatInfinite} \implies (*f* X) N \in \text{HFinite}$
 $\langle \text{proof} \rangle$

lemma *Standard-subset-HFinite*: $\text{Standard} \subseteq \text{HFinite}$
 $\langle \text{proof} \rangle$

lemma *NSBseqD2*: $\text{NSBseq } X \implies (*f* X) N \in \text{HFinite}$
 $\langle \text{proof} \rangle$

lemma *NSBseqI*: $\forall N \in \text{HNatInfinite}. (*f* X) N \in \text{HFinite} \implies \text{NSBseq } X$
 $\langle \text{proof} \rangle$

The standard definition implies the nonstandard definition.

lemma *Bseq-NSBseq*: $\text{Bseq } X \implies \text{NSBseq } X$
 $\langle \text{proof} \rangle$

The nonstandard definition implies the standard definition.

lemma *SReal-less-omega*: $r \in \mathbb{R} \implies r < \omega$
 $\langle \text{proof} \rangle$

lemma *NSBseq-Bseq*: $\text{NSBseq } X \implies \text{Bseq } X$
 $\langle \text{proof} \rangle$

Equivalence of nonstandard and standard definitions for a bounded sequence.

lemma *Bseq-NSBseq-iff*: $\text{Bseq } X = \text{NSBseq } X$
 $\langle \text{proof} \rangle$

A convergent sequence is bounded: Boundedness as a necessary condition for convergence. The nonstandard version has no existential, as usual.

lemma *NSconvergent-NSBseq*: $\text{NSconvergent } X \implies \text{NSBseq } X$
 $\langle \text{proof} \rangle$

Standard Version: easily now proved using equivalence of NS and standard definitions.

lemma *convergent-Bseq*: $\text{convergent } X \implies \text{Bseq } X$
for $X :: \text{nat} \Rightarrow 'b::\text{real-normed-vector}$
 $\langle \text{proof} \rangle$

10.3.1 Upper Bounds and Lubs of Bounded Sequences

lemma *NSBseq-isUb*: $NSBseq\ X \implies \exists U::real. isUb\ UNIV\ \{x. \exists n. X\ n = x\}\ U$
 ⟨proof⟩

lemma *NSBseq-isLub*: $NSBseq\ X \implies \exists U::real. isLub\ UNIV\ \{x. \exists n. X\ n = x\}\ U$
 ⟨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*: $\forall_F\ k\ in\ sequentially. X\ k = X\ m \implies X \longrightarrow_{NS} X\ m$
 ⟨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 $0 \leq x$ and $x < 1$. 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 $(\lambda n. x^n) \longrightarrow_{NS} 0$
 ⟨proof⟩

lemma *NSLIMSEQ-abs-realpow-zero*: $|c| < 1 \implies (\lambda n. |c|^n) \longrightarrow_{NS} 0$
 for $c :: real$
 ⟨proof⟩


```

lemma NSLIMSEQ-abs-realpow-zero2:  $|c| < 1 \implies (\lambda n. c \wedge n) \longrightarrow_{NS} 0$ 
  for  $c :: \text{real}$ 
   $\langle \text{proof} \rangle$ 

end

```

11 Finite Summation and Infinite Series for Hyperreals

```

theory HSeries
  imports HSEQ
begin

```

```

definition sumhr ::  $\text{hypnat} \times \text{hypnat} \times (\text{nat} \Rightarrow \text{real}) \Rightarrow \text{hypreal}$ 
  where  $\text{sumhr} = (\lambda (M, N, f). \text{starfun2 } (\lambda m \ n. \text{sum } f \ \{m..<n\}) \ M \ N)$ 

```

```

definition NSsums ::  $(\text{nat} \Rightarrow \text{real}) \Rightarrow \text{real} \Rightarrow \text{bool}$  (infixr  $\langle \text{NSsums} \rangle$  80)
  where  $f \text{ NSsums } s = (\lambda n. \text{sum } f \ \{..<n\}) \longrightarrow_{NS} s$ 

```

```

definition NSsummable ::  $(\text{nat} \Rightarrow \text{real}) \Rightarrow \text{bool}$ 
  where  $\text{NSsummable } f \longleftrightarrow (\exists s. f \text{ NSsums } s)$ 

```

```

definition NSsuminf ::  $(\text{nat} \Rightarrow \text{real}) \Rightarrow \text{real}$ 
  where  $\text{NSsuminf } f = (\text{THE } s. f \text{ NSsums } s)$ 

```

```

lemma sumhr-app:  $\text{sumhr } (M, N, f) = (*f2* (\lambda m \ n. \text{sum } f \ \{m..<n\})) \ M \ N$ 
   $\langle \text{proof} \rangle$ 

```

Base case in definition of *sumr*.

```

lemma sumhr-zero [simp]:  $\bigwedge m. \text{sumhr } (m, 0, f) = 0$ 
   $\langle \text{proof} \rangle$ 

```

Recursive case in definition of *sumr*.

```

lemma sumhr-if:
   $\bigwedge m \ n. \text{sumhr } (m, n + 1, f) = (\text{if } n + 1 \leq m \text{ then } 0 \text{ else } \text{sumhr } (m, n, f) + (*f* f) \ n)$ 
   $\langle \text{proof} \rangle$ 

```

```

lemma sumhr-Suc-zero [simp]:  $\bigwedge n. \text{sumhr } (n + 1, n, f) = 0$ 
   $\langle \text{proof} \rangle$ 

```

```

lemma sumhr-eq-bounds [simp]:  $\bigwedge n. \text{sumhr } (n, n, f) = 0$ 
   $\langle \text{proof} \rangle$ 

```

```

lemma sumhr-Suc [simp]:  $\bigwedge m. \text{sumhr } (m, m + 1, f) = (*f* f) \ m$ 
   $\langle \text{proof} \rangle$ 

```

lemma *sumhr-add-lbound-zero* [simp]: $\bigwedge k m. \text{sumhr } (m + k, k, f) = 0$
 $\langle \text{proof} \rangle$

lemma *sumhr-add*: $\bigwedge m n. \text{sumhr } (m, n, f) + \text{sumhr } (m, n, g) = \text{sumhr } (m, n, \lambda i. f i + g i)$
 $\langle \text{proof} \rangle$

lemma *sumhr-mult*: $\bigwedge m n. \text{hypreal-of-real } r * \text{sumhr } (m, n, f) = \text{sumhr } (m, n, \lambda n. r * f n)$
 $\langle \text{proof} \rangle$

lemma *sumhr-split-add*: $\bigwedge n p. n < p \implies \text{sumhr } (0, n, f) + \text{sumhr } (n, p, f) = \text{sumhr } (0, p, f)$
 $\langle \text{proof} \rangle$

lemma *sumhr-split-diff*: $n < p \implies \text{sumhr } (0, p, f) - \text{sumhr } (0, n, f) = \text{sumhr } (n, p, f)$
 $\langle \text{proof} \rangle$

lemma *sumhr-hrabs*: $\bigwedge m n. |\text{sumhr } (m, n, f)| \leq \text{sumhr } (m, n, \lambda i. |f i|)$
 $\langle \text{proof} \rangle$

Other general version also needed.

lemma *sumhr-fun-hypnat-eq*:
 $(\forall r. m \leq r \wedge r < n \implies f r = g r) \implies$
 $\text{sumhr } (\text{hypnat-of-nat } m, \text{hypnat-of-nat } n, f) =$
 $\text{sumhr } (\text{hypnat-of-nat } m, \text{hypnat-of-nat } n, g)$
 $\langle \text{proof} \rangle$

lemma *sumhr-const*: $\bigwedge n. \text{sumhr } (0, n, \lambda i. r) = \text{hypreal-of-hypnat } n * \text{hypreal-of-real } r$
 $\langle \text{proof} \rangle$

lemma *sumhr-less-bounds-zero* [simp]: $\bigwedge m n. n < m \implies \text{sumhr } (m, n, f) = 0$
 $\langle \text{proof} \rangle$

lemma *sumhr-minus*: $\bigwedge m n. \text{sumhr } (m, n, \lambda i. -f i) = - \text{sumhr } (m, n, f)$
 $\langle \text{proof} \rangle$

lemma *sumhr-shift-bounds*:
 $\bigwedge m n. \text{sumhr } (m + \text{hypnat-of-nat } k, n + \text{hypnat-of-nat } k, f) =$
 $\text{sumhr } (m, n, \lambda i. f (i + k))$
 $\langle \text{proof} \rangle$

11.1 Nonstandard Sums

Infinite sums are obtained by summing to some infinite hypernatural (such as *whn*).

lemma *sumhr-hypreal-of-hypnat-omega*: $\text{sumhr } (0, \text{whn}, \lambda i. 1) = \text{hypreal-of-hypnat whn}$

<proof>

lemma *whn-eq- ω m1*: $\text{hypreal-of-hypnat whn} = \omega - 1$

<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]: $\bigwedge N. \text{sumhr } (0, N + N, \lambda i. (-1) ^ (i + 1)) = 0$

<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*:

$NSsummable\ f \longleftrightarrow (\forall M \in HNatInfinite. \forall N \in HNatInfinite. |sumhr\ (M, N, f)| \approx 0)$ (is ?L=?R)
 ⟨proof⟩

Terms of a convergent series tend to zero.

lemma *NSsummable-NSLIMSEQ-zero*: $NSsummable\ f \implies f \longrightarrow_{NS} 0$
 ⟨proof⟩

Nonstandard comparison test.

lemma *NSsummable-comparison-test*: $\exists N. \forall n. N \leq n \longrightarrow |f\ n| \leq g\ n \implies NSsummable\ g \implies NSsummable\ f$
 ⟨proof⟩

lemma *NSsummable-rabs-comparison-test*:

$\exists N. \forall n. N \leq n \longrightarrow |f\ n| \leq g\ n \implies NSsummable\ g \implies NSsummable\ (\lambda k. |f\ k|)$
 ⟨proof⟩

end

12 Limits and Continuity (Nonstandard)

theory *HLim*

imports *Star*

abbrevs $----> = -\square\rightarrow_{NS}$

begin

Nonstandard Definitions.

definition *NSLIM* :: $('a::real-normed-vector \Rightarrow 'b::real-normed-vector) \Rightarrow 'a \Rightarrow 'b \Rightarrow bool$

$(\langle \langle notation = \langle mixfix\ NSLIM \rangle \rangle (-) / -(-) / \rightarrow_{NS} (-) \rangle [60, 0, 60]\ 60)$

where $f -a\rightarrow_{NS} L \longleftrightarrow (\forall x. x \neq star-of\ a \wedge x \approx star-of\ a \longrightarrow (*f*\ f)\ x \approx star-of\ L)$

definition *isNSCont* :: $('a::real-normed-vector \Rightarrow 'b::real-normed-vector) \Rightarrow 'a \Rightarrow bool$

where — NS definition dispenses with limit notions

$isNSCont\ f\ a \longleftrightarrow (\forall y. y \approx star-of\ a \longrightarrow (*f*\ f)\ y \approx star-of\ (f\ a))$

definition *isNSUCont* :: $('a::real-normed-vector \Rightarrow 'b::real-normed-vector) \Rightarrow bool$

where $isNSUCont\ f \longleftrightarrow (\forall x\ y. x \approx y \longrightarrow (*f*\ f)\ x \approx (*f*\ f)\ y)$

12.1 Limits of Functions

lemma *NSLIM-I*: $(\bigwedge x. x \neq star-of\ a \implies x \approx star-of\ a \implies starfun\ f\ x \approx star-of\ L) \implies f -a\rightarrow_{NS} L$
 ⟨proof⟩

lemma *NSLIM-D*: $f -a \rightarrow_{NS} L \implies x \neq \text{star-of } a \implies x \approx \text{star-of } a \implies \text{starfun } f \ x \approx \text{star-of } L$
 ⟨proof⟩

Proving properties of limits using nonstandard definition. The properties hold for standard limits as well!

lemma *NSLIM-mult*: $f -x \rightarrow_{NS} l \implies g -x \rightarrow_{NS} m \implies (\lambda x. f \ x * g \ x) -x \rightarrow_{NS} (l * m)$
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*: $f -x \rightarrow_{NS} l \implies g -x \rightarrow_{NS} m \implies (\lambda x. f \ x *_R g \ x) -x \rightarrow_{NS} (l *_R m)$
 ⟨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 $f - x \rightarrow_{NS} l$
 $\langle proof \rangle$

lemma *NSLIM-const-eq*:
fixes $a :: 'a::real-normed-algebra-1$
assumes $(\lambda x. k) - a \rightarrow_{NS} l$
shows $k = l$
 $\langle proof \rangle$

lemma *NSLIM-unique*: $f - a \rightarrow_{NS} l \implies f - a \rightarrow_{NS} M \implies l = M$
for $a :: 'a::real-normed-algebra-1$
 $\langle proof \rangle$

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 $f\ g :: 'a::real-normed-vector \Rightarrow 'b::real-normed-algebra$
 $\langle proof \rangle$

lemma *NSLIM-self*: $(\lambda x. x) - a \rightarrow_{NS} a$
 $\langle proof \rangle$

12.1.1 Equivalence of *filterlim* and *NSLIM*

lemma *LIM-NSLIM*:
assumes $f: f - a \rightarrow L$
shows $f - a \rightarrow_{NS} L$
 $\langle proof \rangle$

lemma *NSLIM-LIM*:
assumes $f: f - a \rightarrow_{NS} L$
shows $f - a \rightarrow L$
 $\langle proof \rangle$

theorem *LIM-NSLIM-iff*: $f - x \rightarrow L \longleftrightarrow f - x \rightarrow_{NS} L$
 $\langle proof \rangle$

12.2 Continuity

lemma *isNSContD*: $isNSCont\ f\ a \implies y \approx star-of\ a \implies (*f* f)\ y \approx star-of\ (f\ a)$
 $\langle proof \rangle$

lemma *isNSCont-NSLIM*: $isNSCont\ f\ a \implies f - a \rightarrow_{NS} (f\ a)$
 $\langle proof \rangle$

lemma *NSLIM-isNSCont*: $f - a \rightarrow_{NS} (f\ a) \implies isNSCont\ f\ a$
 $\langle proof \rangle$

NS continuity can be defined using NS Limit in similar fashion to standard definition of continuity.

lemma *isNSCont-NSLIM-iff*: $\text{isNSCont } f \ a \longleftrightarrow f \ -a \rightarrow_{NS} (f \ a)$
 ⟨proof⟩

Hence, NS continuity can be given in terms of standard limit.

lemma *isNSCont-LIM-iff*: $(\text{isNSCont } f \ a) = (f \ -a \rightarrow (f \ a))$
 ⟨proof⟩

Moreover, it's trivial now that NS continuity is equivalent to standard continuity.

lemma *isNSCont-isCont-iff*: $\text{isNSCont } f \ a \longleftrightarrow \text{isCont } f \ a$
 ⟨proof⟩

Standard continuity \implies NS continuity.

lemma *isCont-isNSCont*: $\text{isCont } f \ a \implies \text{isNSCont } f \ a$
 ⟨proof⟩

NS continuity \implies Standard continuity.

lemma *isNSCont-isCont*: $\text{isNSCont } f \ a \implies \text{isCont } f \ a$
 ⟨proof⟩

Alternative definition of continuity.

Prove equivalence between NS limits – seems easier than using standard definition.

lemma *NSLIM-at0-iff*: $f \ -a \rightarrow_{NS} L \longleftrightarrow (\lambda h. f \ (a + h)) \ -0 \rightarrow_{NS} L$
 ⟨proof⟩

lemma *isNSCont-minus*: $\text{isNSCont } f \ a \implies \text{isNSCont } (\lambda x. - f \ x) \ a$
 ⟨proof⟩

lemma *isNSCont-inverse*: $\text{isNSCont } f \ x \implies f \ x \neq 0 \implies \text{isNSCont } (\lambda x. \text{inverse } (f \ x)) \ x$
for $f :: 'a :: \text{real-normed-vector} \Rightarrow 'b :: \text{real-normed-div-algebra}$
 ⟨proof⟩

lemma *isNSCont-const [simp]*: $\text{isNSCont } (\lambda x. k) \ a$
 ⟨proof⟩

lemma *isNSCont-abs [simp]*: $\text{isNSCont } \text{abs } a$
for $a :: \text{real}$
 ⟨proof⟩

12.3 Uniform Continuity

lemma *isNSUContD*: $\text{isNSUCont } f \implies x \approx y \implies (*f* f) \ x \approx (*f* f) \ y$
 ⟨proof⟩

lemma *isUCont-isNSUCont*:

fixes $f :: 'a::\text{real-normed-vector} \Rightarrow 'b::\text{real-normed-vector}$
assumes $f: \text{isUCont } f$
shows $\text{isNSUCont } f$
 $\langle \text{proof} \rangle$

lemma isNSUCont-isUCont :
fixes $f :: 'a::\text{real-normed-vector} \Rightarrow 'b::\text{real-normed-vector}$
assumes $f: \text{isNSUCont } f$
shows $\text{isUCont } f$
 $\langle \text{proof} \rangle$

end

13 Differentiation (Nonstandard)

theory HDeriv
imports HLim
begin

Nonstandard Definitions.

definition $\text{nsderiv} :: ['a::\text{real-normed-field} \Rightarrow 'a, 'a, 'a] \Rightarrow \text{bool}$
 $(\langle \langle \text{notation} = \langle \text{mixfix } \text{NSDERIV} \rangle \rangle \text{NSDERIV } (-) / (-) / :> (-) \rangle [1000, 1000, 60]$
 $60)$
where $\text{NSDERIV } f \ x :> D \longleftrightarrow$
 $(\forall h \in \text{Infinitesimal} - \{0\}. ((*f* f)(\text{star-of } x + h) - \text{star-of } (f \ x)) / h \approx$
 $\text{star-of } D)$

definition $\text{NSdifferentiable} :: ['a::\text{real-normed-field} \Rightarrow 'a, 'a] \Rightarrow \text{bool}$
 $(\text{infixl } \langle \text{NSdifferentiable} \rangle 60)$
where $f \text{ NSdifferentiable } x \longleftrightarrow (\exists D. \text{NSDERIV } f \ x :> D)$

definition $\text{increment} :: (\text{real} \Rightarrow \text{real}) \Rightarrow \text{real} \Rightarrow \text{hypreal} \Rightarrow \text{hypreal}$
where $\text{increment } f \ x \ h =$
 $(\text{SOME } \text{inc. } f \text{ NSdifferentiable } x \wedge \text{inc} = (*f* f) (\text{hypreal-of-real } x + h) -$
 $\text{hypreal-of-real } (f \ x))$

13.1 Derivatives

lemma DERIV-NS-iff : $(\text{DERIV } f \ x :> D) \longleftrightarrow (\lambda h. (f \ (x + h) - f \ x) / h) - 0 \rightarrow_{NS} D$
 $\langle \text{proof} \rangle$

lemma NS-DERIV-D : $\text{DERIV } f \ x :> D \implies (\lambda h. (f \ (x + h) - f \ x) / h) - 0 \rightarrow_{NS} D$
 $\langle \text{proof} \rangle$

lemma $\text{Infinitesimal-of-hypreal}$:
 $x \in \text{Infinitesimal} \implies ((*f* \text{ of-real } x :: 'a::\text{real-normed-div-algebra } \text{star}) \in \text{Infinitesimal})$

$\langle \text{proof} \rangle$

lemma *of-hypreal-eq-0-iff*: $\bigwedge x. ((\text{*f* of-real } x = (0::'a::\text{real-algebra-1 star})) = (x = 0))$
 $\langle \text{proof} \rangle$

lemma *NSDeriv-unique*:

assumes $\text{NSDERIV } f \ x :> D \ \text{NSDERIV } f \ x :> E$

shows $\text{NSDERIV } f \ x :> D \implies \text{NSDERIV } f \ x :> E \implies D = E$

$\langle \text{proof} \rangle$

First *NSDERIV* in terms of *NSLIM*.

First equivalence.

lemma *NSDERIV-NSLIM-iff*: $(\text{NSDERIV } f \ x :> D) \longleftrightarrow (\lambda h. (f \ (x + h) - f \ x) / h) - 0 \rightarrow_{NS} D$
 $\langle \text{proof} \rangle$

Second equivalence.

lemma *NSDERIV-NSLIM-iff2*: $(\text{NSDERIV } f \ x :> D) \longleftrightarrow (\lambda z. (f \ z - f \ x) / (z - x)) - x \rightarrow_{NS} D$
 $\langle \text{proof} \rangle$

While we're at it!

lemma *NSDERIV-iff2*:

$(\text{NSDERIV } f \ x :> D) \longleftrightarrow$

$(\forall w. w \neq \text{star-of } x \wedge w \approx \text{star-of } x \longrightarrow (\text{*f* } (\lambda z. (f \ z - f \ x) / (z - x))) \ w \approx \text{star-of } D)$

$\langle \text{proof} \rangle$

lemma *NSDERIVD5*:

$\llbracket \text{NSDERIV } f \ x :> D; u \approx \text{hypreal-of-real } x \rrbracket \implies$

$(\text{*f* } (\lambda z. f \ z - f \ x)) \ u \approx \text{hypreal-of-real } D * (u - \text{hypreal-of-real } x)$

$\langle \text{proof} \rangle$

lemma *NSDERIVD4*:

$\llbracket \text{NSDERIV } f \ x :> D; h \in \text{Infinitesimal} \rrbracket$

$\implies (\text{*f* } f)(\text{hypreal-of-real } x + h) - \text{hypreal-of-real } (f \ x) \approx \text{hypreal-of-real } D * h$

$\langle \text{proof} \rangle$

Differentiability implies continuity nice and simple "algebraic" proof.

lemma *NSDERIV-isNSCont*:

assumes $\text{NSDERIV } f \ x :> D$ **shows** $\text{isNSCont } f \ x$

$\langle \text{proof} \rangle$

Differentiation rules for combinations of functions follow from clear, straightforward, 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*:

$\llbracket \text{NSDERIV } g \ x :> D; (\text{*f* } g) (\text{star-of } x + y) = \text{star-of } (g \ x); y \in \text{Infinitesimal}; y \neq 0 \rrbracket$
 $\implies D = 0$
 $\langle \text{proof} \rangle$

Can be proved differently using *NSLIM-isCont-iff*.

lemma *NSDERIV-approx*:

$\text{NSDERIV } f \ x :> D \implies h \in \text{Infinitesimal} \implies h \neq 0 \implies$
 $(\text{*f* } f) (\text{star-of } x + h) - \text{star-of } (f \ x) \approx 0$
 $\langle \text{proof} \rangle$

From one version of differentiability

$$f \ x - f \ a \text{ ----- } \approx D b \ x - a$$

lemma *NSDERIVD1*:

$\llbracket \text{NSDERIV } f \ (g \ x) :> D a;$
 $(\text{*f* } g) (\text{star-of } x + y) \neq \text{star-of } (g \ x);$
 $(\text{*f* } g) (\text{star-of } x + y) \approx \text{star-of } (g \ x) \rrbracket$
 $\implies ((\text{*f* } f) ((\text{*f* } g) (\text{star-of } x + y)) -$
 $\text{star-of } (f \ (g \ x))) / ((\text{*f* } g) (\text{star-of } x + y) - \text{star-of } (g \ x)) \approx$
 $\text{star-of } D a$
 $\langle \text{proof} \rangle$

From other version of differentiability

$$f \ (x + h) - f \ x \text{ ----- } \approx D b \ h$$

lemma *NSDERIVD2*: $\llbracket \text{NSDERIV } g \ x :> D b; y \in \text{Infinitesimal}; y \neq 0 \rrbracket$
 $\implies ((\text{*f* } g) (\text{star-of } (x) + y) - \text{star-of } (g \ x)) / y$
 $\approx \text{star-of } (D b)$
 $\langle \text{proof} \rangle$

This proof uses both definitions of differentiability.

lemma *NSDERIV-chain*:

$\text{NSDERIV } f \ (g \ x) :> D a \implies \text{NSDERIV } g \ x :> D b \implies \text{NSDERIV } (f \circ g) \ x :>$
 $D a * D b$
 $\langle \text{proof} \rangle$

Differentiation of natural number powers.

lemma *NSDERIV-Id [simp]*: $\text{NSDERIV } (\lambda x. x) \ x :> 1$
 $\langle \text{proof} \rangle$

lemma *NSDERIV-cmult-Id [simp]*: $\text{NSDERIV } ((*) \ c) \ x :> c$
 $\langle \text{proof} \rangle$

lemma *NSDERIV-inverse*:

fixes $x :: 'a::\text{real-normed-field}$
assumes $x \neq 0$ — can't get rid of $x \neq 0$ because it isn't continuous at zero
shows $\text{NSDERIV } (\lambda x. \text{inverse } x) \ x :> - (\text{inverse } x \wedge \text{Suc } (\text{Suc } 0))$
 $\langle \text{proof} \rangle$

13.2.1 Equivalence of NS and Standard definitions

lemma *divideR-eq-divide*: $x \text{ / }_R y = x \text{ / } y$
 $\langle \text{proof} \rangle$

Now equivalence between *NSDERIV* and *DERIV*.

lemma *NSDERIV-DERIV-iff*: $\text{NSDERIV } f \ x :> D \longleftrightarrow \text{DERIV } f \ x :> D$
 $\langle \text{proof} \rangle$

NS version.

lemma *NSDERIV-pow*: $\text{NSDERIV } (\lambda x. x \wedge^n) \ x :> \text{real } n * (x \wedge (n - \text{Suc } 0))$
 $\langle \text{proof} \rangle$

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 $x :: 'a :: \{\text{real-normed-field}\}$
 $\langle \text{proof} \rangle$

Derivative of quotient.

lemma *NSDERIV-quotient*:
fixes $x :: 'a :: \text{real-normed-field}$
shows $\text{NSDERIV } f \ x :> d \implies \text{NSDERIV } g \ x :> e \implies g \ x \neq 0 \implies$
 $\text{NSDERIV } (\lambda y. f \ y / g \ y) \ x :> (d * g \ x - (e * f \ x)) / (g \ x \wedge \text{Suc } (\text{Suc } 0))$
 $\langle \text{proof} \rangle$

lemma *CARAT-NSDERIV*:
 $\text{NSDERIV } f \ x :> l \implies \exists g. (\forall z. f \ z - f \ x = g \ z * (z - x)) \wedge \text{isNSCont } g \ x \wedge g$
 $x = l$
 $\langle \text{proof} \rangle$

lemma *hypreal-eq-minus-iff3*: $x = y + z \longleftrightarrow x + - z = y$
for $x \ y \ z :: \text{hypreal}$
 $\langle \text{proof} \rangle$

lemma *CARAT-DERIVD*:
assumes $\text{all}: \forall z. f \ z - f \ x = g \ z * (z - x)$
and $\text{nsc}: \text{isNSCont } g \ x$
shows $\text{NSDERIV } f \ x :> g \ x$
 $\langle \text{proof} \rangle$

13.2.2 Differentiability predicate

lemma *NSdifferentiableD*: $f \ \text{NSdifferentiable } x \implies \exists D. \text{NSDERIV } f \ x :> D$
 $\langle \text{proof} \rangle$

lemma *NSdifferentiableI*: $\text{NSDERIV } f \ x :> D \implies f \ \text{NSdifferentiable } x$
 $\langle \text{proof} \rangle$

13.3 (NS) Increment

lemma *incrementI1*:

f NSdifferentiable x \implies

increment f x h = (**f** *f*) (*hypreal-of-real x + h*) – *hypreal-of-real (f x)*

<proof>

lemma *incrementI2*:

NSDERIV f x :> D \implies

increment f x h = (**f** *f*) (*hypreal-of-real x + h*) – *hypreal-of-real (f x)*

<proof>

The Increment theorem – Keisler p. 65.

lemma *increment-thm*:

assumes *NSDERIV f x :> D* *h* \in *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 ≈ 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 $\equiv st(\text{sum hr } (0, \text{whn}, \lambda n. \text{inverse } (\text{fact } n) * (x \wedge n)))$

definition

sin hr :: *real* \Rightarrow *hypreal* **where**

sin hr x $\equiv st(\text{sum hr } (0, \text{whn}, \lambda n. \text{sin-coeff } n * x \wedge n))$

definition

cosh hr :: *real* \Rightarrow *hypreal* **where**

cosh hr x $\equiv st(\text{sum hr } (0, \text{whn}, \lambda n. \text{cos-coeff } n * x \wedge n))$

14.1 Nonstandard Extension of Square Root Function

lemma *STAR-sqrt-zero [simp]*: (**f** *sqrt*) 0 = 0

<proof>

lemma *STAR-sqrt-one* [simp]: $(\text{*f* sqrt})\ 1 = 1$
 ⟨proof⟩

lemma *hypreal-sqrt-pow2-iff*: $((\text{*f* sqrt})(x))^2 = x = (0 \leq x)$
 ⟨proof⟩

lemma *hypreal-sqrt-gt-zero-pow2*: $\bigwedge x. 0 < x \implies (\text{*f* sqrt})\ x)^2 = x$
 ⟨proof⟩

lemma *hypreal-sqrt-pow2-gt-zero*: $0 < x \implies 0 < (\text{*f* sqrt})\ x)^2$
 ⟨proof⟩

lemma *hypreal-sqrt-not-zero*: $0 < x \implies (\text{*f* sqrt})\ x \neq 0$
 ⟨proof⟩

lemma *hypreal-inverse-sqrt-pow2*:
 $0 < x \implies \text{inverse } ((\text{*f* sqrt})(x))^2 = \text{inverse } x$
 ⟨proof⟩

lemma *hypreal-sqrt-mult-distrib*:
 $\bigwedge x\ y. \llbracket 0 < x; 0 < y \rrbracket \implies$
 $(\text{*f* sqrt})(x*y) = (\text{*f* sqrt})(x) * (\text{*f* sqrt})(y)$
 ⟨proof⟩

lemma *hypreal-sqrt-mult-distrib2*:
 $\llbracket 0 \leq x; 0 \leq y \rrbracket \implies (\text{*f* sqrt})(x*y) = (\text{*f* sqrt})(x) * (\text{*f* sqrt})(y)$
 ⟨proof⟩

lemma *hypreal-sqrt-approx-zero* [simp]:
 assumes $0 < x$
 shows $((\text{*f* sqrt})\ x \approx 0) \longleftrightarrow (x \approx 0)$
 ⟨proof⟩

lemma *hypreal-sqrt-approx-zero2* [simp]:
 $0 \leq x \implies ((\text{*f* sqrt})(x) \approx 0) = (x \approx 0)$
 ⟨proof⟩

lemma *hypreal-sqrt-gt-zero*: $\bigwedge x. 0 < x \implies 0 < (\text{*f* sqrt})(x)$
 ⟨proof⟩

lemma *hypreal-sqrt-ge-zero*: $0 \leq x \implies 0 \leq (\text{*f* sqrt})(x)$
 ⟨proof⟩

lemma *hypreal-sqrt-lessI*:
 $\bigwedge x\ u. \llbracket 0 < u; x < u^2 \rrbracket \implies (\text{*f* sqrt})\ x < u$
 ⟨proof⟩

lemma *hypreal-sqrt-hrabs* [simp]: $\bigwedge x. (\text{*f* sqrt})(x^2) = |x|$
 ⟨proof⟩

lemma *hypreal-sqrt-hrabs2* [simp]: $\bigwedge x. (*f* \text{ sqrt})(x*x) = |x|$
 ⟨proof⟩

lemma *hypreal-sqrt-hyperpow-hrabs* [simp]:
 $\bigwedge x. (*f* \text{ sqrt})(x \text{ pow } (\text{hypnat-of-nat } 2)) = |x|$
 ⟨proof⟩

lemma *star-sqrt-HFinite*: $\llbracket x \in \text{HFinite}; 0 \leq x \rrbracket \implies (*f* \text{ sqrt}) x \in \text{HFinite}$
 ⟨proof⟩

lemma *st-hypreal-sqrt*:
 assumes $x \in \text{HFinite}$ $0 \leq x$
 shows $st((*f* \text{ sqrt}) x) = (*f* \text{ sqrt})(st x)$
 ⟨proof⟩

lemma *hypreal-sqrt-sum-squares-ge1* [simp]: $\bigwedge x y. x \leq (*f* \text{ sqrt})(x^2 + y^2)$
 ⟨proof⟩

lemma *HFinite-hypreal-sqrt-imp-HFinite*:
 $\llbracket 0 \leq x; (*f* \text{ sqrt}) x \in \text{HFinite} \rrbracket \implies x \in \text{HFinite}$
 ⟨proof⟩

lemma *HFinite-hypreal-sqrt-iff* [simp]:
 $0 \leq x \implies ((*f* \text{ sqrt}) x \in \text{HFinite}) = (x \in \text{HFinite})$
 ⟨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]:
 $0 \leq x \implies ((*f* \text{ sqrt}) x \in \text{HInfinite}) = (x \in \text{HInfinite})$

$\langle \text{proof} \rangle$

lemma *HFinite-exp* [simp]:
 $\text{sumhr } (0, \text{whn}, \lambda n. \text{inverse } (\text{fact } n) * x ^ n) \in \text{HFinite}$
 $\langle \text{proof} \rangle$

lemma *exp-hr-zero* [simp]: $\text{exp-hr } 0 = 1$
 $\langle \text{proof} \rangle$

lemma *cosh-hr-zero* [simp]: $\text{cosh-hr } 0 = 1$
 $\langle \text{proof} \rangle$

lemma *STAR-exp-zero-approx-one* [simp]: $(*f* \text{ exp}) (0::\text{hypreal}) \approx 1$
 $\langle \text{proof} \rangle$

lemma *STAR-exp-Infinitesimal*:
assumes $x \in \text{Infinitesimal}$ **shows** $(*f* \text{ exp}) (x::\text{hypreal}) \approx 1$
 $\langle \text{proof} \rangle$

lemma *STAR-exp-epsilon* [simp]: $(*f* \text{ exp}) \varepsilon \approx 1$
 $\langle \text{proof} \rangle$

lemma *STAR-exp-add*:
 $\bigwedge (x::'a::\{\text{banach,real-normed-field}\} \text{ star}) y. (*f* \text{ exp})(x + y) = (*f* \text{ exp}) x * (*f* \text{ exp}) y$
 $\langle \text{proof} \rangle$

lemma *exp-hr-hypreal-of-real-exp-eq*: $\text{exp-hr } x = \text{hypreal-of-real } (\text{exp } x)$
 $\langle \text{proof} \rangle$

lemma *starfun-exp-ge-add-one-self* [simp]: $\bigwedge x::\text{hypreal}. 0 \leq x \implies (1 + x) \leq (*f* \text{ exp}) x$
 $\langle \text{proof} \rangle$

exp maps infinities to infinities

lemma *starfun-exp-HInfinite*:
fixes $x :: \text{hypreal}$
assumes $x \in \text{HInfinite}$ $0 \leq x$
shows $(*f* \text{ exp}) x \in \text{HInfinite}$
 $\langle \text{proof} \rangle$

lemma *starfun-exp-minus*:
 $\bigwedge (x::'a::\{\text{banach,real-normed-field}\} \text{ star}). (*f* \text{ exp}) (-x) = \text{inverse}((*f* \text{ exp}) x)$
 $\langle \text{proof} \rangle$

exp maps infinitesimals to infinitesimals

lemma *starfun-exp-Infinitesimal*:
fixes $x :: \text{hypreal}$
assumes $x \in \text{HInfinite}$ $x \leq 0$

shows $(*f* \exp) x \in \text{Infinitesimal}$
 $\langle \text{proof} \rangle$

lemma *starfun-exp-gt-one* [simp]: $\bigwedge x::\text{hypreal}. 0 < x \implies 1 < (*f* \exp) x$
 $\langle \text{proof} \rangle$

abbreviation *real-ln* :: $\text{real} \Rightarrow \text{real}$ **where**
 $\text{real-ln} \equiv \text{ln}$

lemma *starfun-ln-exp* [simp]: $\bigwedge x. (*f* \text{real-ln}) ((*f* \exp) x) = x$
 $\langle \text{proof} \rangle$

lemma *starfun-exp-ln-iff* [simp]: $\bigwedge x. ((*f* \exp)((*f* \text{real-ln}) x) = x) = (0 < x)$
 $\langle \text{proof} \rangle$

lemma *starfun-exp-ln-eq*: $\bigwedge u x. (*f* \exp) u = x \implies (*f* \text{real-ln}) x = u$
 $\langle \text{proof} \rangle$

lemma *starfun-ln-less-self* [simp]: $\bigwedge x. 0 < x \implies (*f* \text{real-ln}) x < x$
 $\langle \text{proof} \rangle$

lemma *starfun-ln-ge-zero* [simp]: $\bigwedge x. 1 \leq x \implies 0 \leq (*f* \text{real-ln}) x$
 $\langle \text{proof} \rangle$

lemma *starfun-ln-gt-zero* [simp]: $\bigwedge x. 1 < x \implies 0 < (*f* \text{real-ln}) x$
 $\langle \text{proof} \rangle$

lemma *starfun-ln-not-eq-zero* [simp]: $\bigwedge x. \llbracket 0 < x; x \neq 1 \rrbracket \implies (*f* \text{real-ln}) x \neq 0$
 $\langle \text{proof} \rangle$

lemma *starfun-ln-HFinite*: $\llbracket x \in \text{HFinite}; 1 \leq x \rrbracket \implies (*f* \text{real-ln}) x \in \text{HFinite}$
 $\langle \text{proof} \rangle$

lemma *starfun-ln-inverse*: $\bigwedge x. 0 < x \implies (*f* \text{real-ln}) (\text{inverse } x) = -(*f* \text{ln}) x$
 $\langle \text{proof} \rangle$

lemma *starfun-abs-exp-cancel*: $\bigwedge x. |(*f* \exp) (x::\text{hypreal})| = (*f* \exp) x$
 $\langle \text{proof} \rangle$

lemma *starfun-exp-less-mono*: $\bigwedge x y::\text{hypreal}. x < y \implies (*f* \exp) x < (*f* \exp) y$
 $\langle \text{proof} \rangle$

lemma *starfun-exp-HFinite*:
fixes $x::\text{hypreal}$
assumes $x \in \text{HFinite}$
shows $(*f* \exp) x \in \text{HFinite}$
 $\langle \text{proof} \rangle$

lemma *starfun-exp-add-HFinite-Infinitesimal-approx*:

fixes $x :: \text{hypreal}$
shows $\llbracket x \in \text{Infinitesimal}; z \in \text{HFinite} \rrbracket \implies (*f* \exp) (z + x::\text{hypreal}) \approx (*f* \exp) z$
 $\langle \text{proof} \rangle$

lemma *starfun-ln-HInfinite*:

shows $\llbracket x \in \text{HInfinite}; 0 < x \rrbracket \implies (*f* \text{real-ln}) x \in \text{HInfinite}$
 $\langle \text{proof} \rangle$

lemma *starfun-exp-HInfinite-Infinitesimal-disj*:

fixes $x :: \text{hypreal}$
shows $x \in \text{HInfinite} \implies (*f* \exp) x \in \text{HInfinite} \vee (*f* \exp) (x::\text{hypreal}) \in \text{Infinitesimal}$
 $\langle \text{proof} \rangle$

lemma *starfun-ln-HFinite-not-Infinitesimal*:

shows $\llbracket x \in \text{HFinite} - \text{Infinitesimal}; 0 < x \rrbracket \implies (*f* \text{real-ln}) x \in \text{HFinite}$
 $\langle \text{proof} \rangle$

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*:

shows $\llbracket x \in \text{Infinitesimal}; 0 < x \rrbracket \implies (*f* \text{real-ln}) x < 0$
 $\langle \text{proof} \rangle$

lemma *starfun-ln-HInfinite-gt-zero*:

shows $\llbracket x \in \text{HInfinite}; 0 < x \rrbracket \implies 0 < (*f* \text{real-ln}) x$
 $\langle \text{proof} \rangle$

lemma *HFinite-sin [simp]*: $\text{sumhr } (0, \text{whn}, \lambda n. \text{sin-coeff } n * x ^ n) \in \text{HFinite}$

$\langle \text{proof} \rangle$

lemma *STAR-sin-zero [simp]*: $(*f* \text{sin}) 0 = 0$

$\langle \text{proof} \rangle$

lemma *STAR-sin-Infinitesimal [simp]*:

fixes $x :: 'a::\{\text{real-normed-field}, \text{banach}\}$ *star*
assumes $x \in \text{Infinitesimal}$
shows $(*f* \text{sin}) x \approx x$

$\langle \text{proof} \rangle$

lemma *HFinite-cos* [simp]: $\text{sumhr } (0, \text{whn}, \lambda n. \text{cos-coeff } n * x ^ n) \in \text{HFinite}$
 $\langle \text{proof} \rangle$

lemma *STAR-cos-zero* [simp]: $(*f* \text{cos}) 0 = 1$
 $\langle \text{proof} \rangle$

lemma *STAR-cos-Infinitesimal* [simp]:
fixes $x :: 'a :: \{\text{real-normed-field}, \text{banach}\}$ *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 $(*f* \text{tan}) x \approx x$
 $\langle \text{proof} \rangle$

lemma *STAR-sin-cos-Infinitesimal-mult*:
fixes $x :: 'a :: \{\text{real-normed-field}, \text{banach}\}$ *star*
shows $x \in \text{Infinitesimal} \implies (*f* \text{sin}) x * (*f* \text{cos}) x \approx x$
 $\langle \text{proof} \rangle$

lemma *HFinite-pi*: $\text{hypreal-of-real } \pi \in \text{HFinite}$
 $\langle \text{proof} \rangle$

lemma *STAR-sin-Infinitesimal-divide*:
fixes $x :: 'a :: \{\text{real-normed-field}, \text{banach}\}$ *star*
shows $\llbracket x \in \text{Infinitesimal}; x \neq 0 \rrbracket \implies (*f* \text{sin}) x / x \approx 1$
 $\langle \text{proof} \rangle$

14.2 Proving $\sin * (1/n) \times 1/(1/n) \approx 1$ for $n = \infty$

lemma *lemma-sin-pi*:
 $n \in \text{HNatInfinite}$
 $\implies (*f* \text{sin}) (\text{inverse } (\text{hypreal-of-hypnat } n)) / (\text{inverse } (\text{hypreal-of-hypnat } n))$
 ≈ 1
 $\langle \text{proof} \rangle$

lemma *STAR-sin-inverse-HNatInfinite*:
 $n \in \text{HNatInfinite}$
 $\implies (*f* \text{sin}) (\text{inverse } (\text{hypreal-of-hypnat } n)) * \text{hypreal-of-hypnat } n \approx 1$
 $\langle \text{proof} \rangle$

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 $(\ast f \ast \sin) (\text{hypreal-of-real } \pi / \text{hypreal-of-hypnat } n) \ast \text{hypreal-of-hypnat } n$

\approx

$\text{hypreal-of-real } \pi$

$\langle \text{proof} \rangle$

lemma *STAR-sin-pi-divide-HNatInfinite-approx-pi2*:

$n \in \text{HNatInfinite}$

$\implies \text{hypreal-of-hypnat } n \ast (\ast f \ast \sin) (\text{hypreal-of-real } \pi / (\text{hypreal-of-hypnat } n))$

$\approx \text{hypreal-of-real } \pi$

$\langle \text{proof} \rangle$

lemma *starfunNat-pi-divide-n-Infinitesimal*:

$N \in \text{HNatInfinite} \implies (\ast f \ast (\lambda x. \pi / \text{real } x)) N \in \text{Infinitesimal}$

$\langle \text{proof} \rangle$

lemma *STAR-sin-pi-divide-n-approx*:

assumes $N \in \text{HNatInfinite}$

shows $(\ast f \ast \sin) ((\ast f \ast (\lambda x. \pi / \text{real } x)) N) \approx \text{hypreal-of-real } \pi / (\text{hypreal-of-hypnat } N)$

$\langle \text{proof} \rangle$

lemma *NSLIMSEQ-sin-pi*: $(\lambda n. \text{real } n \ast \sin (\pi / \text{real } n)) \longrightarrow_{NS} \pi$

$\langle \text{proof} \rangle$

lemma *NSLIMSEQ-cos-one*: $(\lambda n. \cos (\pi / \text{real } n)) \longrightarrow_{NS} 1$

$\langle \text{proof} \rangle$

lemma *NSLIMSEQ-sin-cos-pi*:

$(\lambda n. \text{real } n \ast \sin (\pi / \text{real } n) \ast \cos (\pi / \text{real } n)) \longrightarrow_{NS} \pi$

$\langle \text{proof} \rangle$

A familiar approximation to $\cos x$ when x is small

lemma *STAR-cos-Infinitesimal-approx*:

fixes $x :: 'a :: \{\text{real-normed-field}, \text{banach}\}$ *star*

shows $x \in \text{Infinitesimal} \implies (\ast f \ast \cos) x \approx 1 - x^2$

$\langle \text{proof} \rangle$

lemma *STAR-cos-Infinitesimal-approx2*:

```

fixes  $x :: \text{hypreal}$ 
assumes  $x \in \text{Infinitesimal}$ 
shows  $(\text{*f* cos}) x \approx 1 - (x^2)/2$ 
 $\langle \text{proof} \rangle$ 

end

```

15 Non-Standard Complex Analysis

```

theory NSCA
imports NSComplex HTranscendental
begin

```

abbreviation

```

 $SComplex :: \text{hcomplex set}$  where
 $SComplex \equiv \text{Standard}$ 

```

definition — standard part map

```

 $stc :: \text{hcomplex} \Rightarrow \text{hcomplex}$  where
 $stc\ x = (\text{SOME } r. x \in \text{HFinite} \wedge r \in SComplex \wedge r \approx x)$ 

```

15.1 Closure Laws for SComplex, the Standard Complex Numbers

lemma *SComplex-minus-iff* [simp]: $(-x \in SComplex) = (x \in SComplex)$
 $\langle \text{proof} \rangle$

lemma *SComplex-add-cancel*:

```

 $\llbracket x + y \in SComplex; y \in SComplex \rrbracket \Longrightarrow x \in SComplex$ 
 $\langle \text{proof} \rangle$ 

```

lemma *SReal-hcmod-hcomplex-of-complex* [simp]:

```

 $hcmod\ (\text{hcomplex-of-complex } r) \in \mathbb{R}$ 
 $\langle \text{proof} \rangle$ 

```

lemma *SReal-hcmod-numeral*: $hcmod\ (\text{numeral } w :: \text{hcomplex}) \in \mathbb{R}$

$\langle \text{proof} \rangle$

lemma *SReal-hcmod-SComplex*: $x \in SComplex \Longrightarrow hcmod\ x \in \mathbb{R}$

$\langle \text{proof} \rangle$

lemma *SComplex-divide-numeral*:

```

 $r \in SComplex \Longrightarrow r / (\text{numeral } w :: \text{hcomplex}) \in SComplex$ 
 $\langle \text{proof} \rangle$ 

```

lemma *SComplex-UNIV-complex*:

```

 $\{x. \text{hcomplex-of-complex } x \in SComplex\} = (\text{UNIV} :: \text{complex set})$ 
 $\langle \text{proof} \rangle$ 

```

lemma *SComplex-iff*: $(x \in SComplex) = (\exists y. x = hcomplex-of-complex y)$
 $\langle proof \rangle$

lemma *hcomplex-of-complex-image*:
 $range\ hcomplex-of-complex = SComplex$
 $\langle proof \rangle$

lemma *inv-hcomplex-of-complex-image*: $inv\ hcomplex-of-complex\ 'SComplex = UNIV$
 $\langle proof \rangle$

lemma *SComplex-hcomplex-of-complex-image*:
 $\llbracket \exists x. x \in P; P \leq SComplex \rrbracket \implies \exists Q. P = hcomplex-of-complex\ 'Q$
 $\langle proof \rangle$

lemma *SComplex-SReal-dense*:
 $\llbracket x \in SComplex; y \in SComplex; hmod\ x < hmod\ y \rrbracket \implies \exists r \in Reals. hmod\ x < r \wedge r < hmod\ y$
 $\langle proof \rangle$

15.2 The Finite Elements form a Subring

lemma *HFinite-hmod-hcomplex-of-complex [simp]*:
 $hmod\ (hcomplex-of-complex\ r) \in HFinite$
 $\langle proof \rangle$

lemma *HFinite-hmod-iff [simp]*: $hmod\ x \in HFinite \longleftrightarrow x \in HFinite$
 $\langle proof \rangle$

lemma *HFinite-bounded-hmod*:
 $\llbracket x \in HFinite; y \leq hmod\ x; 0 \leq y \rrbracket \implies y \in HFinite$
 $\langle proof \rangle$

15.3 The Complex Infinitesimals form a Subring

lemma *Infinitesimal-hmod-iff*:
 $(z \in Infinitesimal) = (hmod\ z \in Infinitesimal)$
 $\langle proof \rangle$

lemma *HInfinite-hmod-iff*: $(z \in HInfinite) = (hmod\ z \in HInfinite)$
 $\langle proof \rangle$

lemma *HFinite-diff-Infinitesimal-hmod*:
 $x \in HFinite - Infinitesimal \implies hmod\ x \in HFinite - Infinitesimal$
 $\langle proof \rangle$

lemma *hmod-less-Infinitesimal*:
 $\llbracket e \in Infinitesimal; hmod\ x < hmod\ e \rrbracket \implies x \in Infinitesimal$
 $\langle proof \rangle$

lemma *hcmmod-le-Infinitesimal*:

$$\llbracket e \in \text{Infinitesimal}; \text{hcmmod } x \leq \text{hcmmod } e \rrbracket \implies x \in \text{Infinitesimal}$$

<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]:

$$\llbracket a \in \text{SComplex}; a \neq 0 \rrbracket \implies (a*x \approx 0) = (x \approx 0)$$

<proof>

lemma *approx-SComplex-mult-cancel*:

$$\llbracket a \in \text{SComplex}; a \neq 0; a*w \approx a*z \rrbracket \implies w \approx z$$

<proof>

lemma *approx-SComplex-mult-cancel-iff1* [simp]:

$$\llbracket a \in \text{SComplex}; a \neq 0 \rrbracket \implies (a*w \approx a*z) = (w \approx z)$$

<proof>

lemma *approx-hcmmod-approx-zero*: $(x \approx y) = (\text{hcmmod } (y - x) \approx 0)$
<proof>

lemma *approx-approx-zero-iff*: $(x \approx 0) = (\text{hcmmod } x \approx 0)$
<proof>

lemma *approx-minus-zero-cancel-iff* [simp]: $(-x \approx 0) = (x \approx 0)$
<proof>

lemma *Infinitesimal-hcmmod-add-diff*:

$$u \approx 0 \implies \text{hcmmod}(x + u) - \text{hcmmod } x \in \text{Infinitesimal}$$

<proof>

lemma *approx-hcmmod-add-hcmmod*: $u \approx 0 \implies \text{hcmmod}(x + u) \approx \text{hcmmod } x$
<proof>

15.5 Zero is the Only Infinitesimal Complex Number

lemma *Infinitesimal-less-SComplex*:

$$\llbracket x \in \text{SComplex}; y \in \text{Infinitesimal}; 0 < \text{hcmmod } x \rrbracket \implies \text{hcmmod } y < \text{hcmmod } x$$

$\langle \text{proof} \rangle$

lemma *SComplex-Int-Infinesimal-zero*: $SComplex\ Int\ Infinitesimal = \{0\}$
 $\langle \text{proof} \rangle$

lemma *SComplex-Infinesimal-zero*:
 $\llbracket x \in SComplex; x \in Infinitesimal \rrbracket \implies x = 0$
 $\langle \text{proof} \rangle$

lemma *SComplex-HFinite-diff-Infinesimal*:
 $\llbracket x \in SComplex; x \neq 0 \rrbracket \implies x \in HFinite - Infinitesimal$
 $\langle \text{proof} \rangle$

lemma *numeral-not-Infinesimal* [simp]:
 $numeral\ w \neq (0::hcomplex) \implies (numeral\ w::hcomplex) \notin Infinitesimal$
 $\langle \text{proof} \rangle$

lemma *approx-SComplex-not-zero*:
 $\llbracket y \in SComplex; x \approx y; y \neq 0 \rrbracket \implies x \neq 0$
 $\langle \text{proof} \rangle$

lemma *SComplex-approx-iff*:
 $\llbracket x \in SComplex; y \in SComplex \rrbracket \implies (x \approx y) = (x = y)$
 $\langle \text{proof} \rangle$

lemma *approx-unique-complex*:
 $\llbracket r \in SComplex; s \in SComplex; r \approx x; s \approx x \rrbracket \implies r = s$
 $\langle \text{proof} \rangle$

15.6 Properties of hRe , hIm and $HComplex$

lemma *abs-hRe-le-hcmod*: $\bigwedge x. |hRe\ x| \leq hcmod\ x$
 $\langle \text{proof} \rangle$

lemma *abs-hIm-le-hcmod*: $\bigwedge x. |hIm\ x| \leq hcmod\ x$
 $\langle \text{proof} \rangle$

lemma *Infinitesimal-hRe*: $x \in Infinitesimal \implies hRe\ x \in Infinitesimal$
 $\langle \text{proof} \rangle$

lemma *Infinitesimal-hIm*: $x \in Infinitesimal \implies hIm\ x \in Infinitesimal$
 $\langle \text{proof} \rangle$

lemma *Infinitesimal-HComplex*:
assumes $x: x \in Infinitesimal$ **and** $y: y \in Infinitesimal$
shows $HComplex\ x\ y \in Infinitesimal$
 $\langle \text{proof} \rangle$

lemma *hcomplex-Infinitesimal-iff*:

$(x \in \text{Infinitesimal}) \longleftrightarrow (hRe\ x \in \text{Infinitesimal} \wedge hIm\ x \in \text{Infinitesimal})$
 $\langle \text{proof} \rangle$

lemma *hRe-diff* [simp]: $\bigwedge x\ y. hRe\ (x - y) = hRe\ x - hRe\ y$
 $\langle \text{proof} \rangle$

lemma *hIm-diff* [simp]: $\bigwedge x\ y. hIm\ (x - y) = hIm\ x - hIm\ y$
 $\langle \text{proof} \rangle$

lemma *approx-hRe*: $x \approx y \implies hRe\ x \approx hRe\ y$
 $\langle \text{proof} \rangle$

lemma *approx-hIm*: $x \approx y \implies hIm\ x \approx hIm\ y$
 $\langle \text{proof} \rangle$

lemma *approx-HComplex*:
 $\llbracket a \approx b; c \approx d \rrbracket \implies HComplex\ a\ c \approx HComplex\ b\ d$
 $\langle \text{proof} \rangle$

lemma *hcomplex-approx-iff*:
 $(x \approx y) = (hRe\ x \approx hRe\ y \wedge hIm\ x \approx hIm\ y)$
 $\langle \text{proof} \rangle$

lemma *HFinite-hRe*: $x \in HFinite \implies hRe\ x \in HFinite$
 $\langle \text{proof} \rangle$

lemma *HFinite-hIm*: $x \in HFinite \implies hIm\ x \in HFinite$
 $\langle \text{proof} \rangle$

lemma *HFinite-HComplex*:
assumes $x \in HFinite\ y \in HFinite$
shows $HComplex\ x\ y \in HFinite$
 $\langle \text{proof} \rangle$

lemma *hcomplex-HFinite-iff*:
 $(x \in HFinite) = (hRe\ x \in HFinite \wedge hIm\ x \in HFinite)$
 $\langle \text{proof} \rangle$

lemma *hcomplex-HInfinite-iff*:
 $(x \in HInfinite) = (hRe\ x \in HInfinite \vee hIm\ x \in HInfinite)$
 $\langle \text{proof} \rangle$

lemma *hcomplex-of-hypreal-approx-iff* [simp]:
 $(hcomplex\ of\ hypreal\ x \approx hcomplex\ of\ hypreal\ z) = (x \approx z)$
 $\langle \text{proof} \rangle$

lemma *stc-part-Ex*:
assumes $x \in HFinite$

shows $\exists t \in SComplex. x \approx t$
 $\langle proof \rangle$

lemma *stc-part-Ex1*: $x \in HFinite \implies \exists! t. t \in SComplex \wedge x \approx t$
 $\langle proof \rangle$

15.7 Theorems About Monads

lemma *monad-zero-hcmo-iff*: $(x \in monad\ 0) = (hcmo\ x \in monad\ 0)$
 $\langle proof \rangle$

15.8 Theorems About Standard Part

lemma *stc-approx-self*: $x \in HFinite \implies stc\ x \approx x$
 $\langle proof \rangle$

lemma *stc-SComplex*: $x \in HFinite \implies stc\ x \in SComplex$
 $\langle proof \rangle$

lemma *stc-HFinite*: $x \in HFinite \implies stc\ x \in HFinite$
 $\langle proof \rangle$

lemma *stc-unique*: $\llbracket y \in SComplex; y \approx x \rrbracket \implies stc\ x = y$
 $\langle proof \rangle$

lemma *stc-SComplex-eq [simp]*: $x \in SComplex \implies stc\ x = x$
 $\langle proof \rangle$

lemma *stc-eq-approx*:
 $\llbracket x \in HFinite; y \in HFinite; stc\ x = stc\ y \rrbracket \implies x \approx y$
 $\langle proof \rangle$

lemma *approx-stc-eq*:
 $\llbracket x \in HFinite; y \in HFinite; x \approx y \rrbracket \implies stc\ x = stc\ y$
 $\langle proof \rangle$

lemma *stc-eq-approx-iff*:
 $\llbracket x \in HFinite; y \in HFinite \rrbracket \implies (x \approx y) = (stc\ x = stc\ y)$
 $\langle proof \rangle$

lemma *stc-Infinitesimal-add-SComplex*:
 $\llbracket x \in SComplex; e \in Infinitesimal \rrbracket \implies stc(x + e) = x$
 $\langle proof \rangle$

lemma *stc-Infinitesimal-add-SComplex2*:
 $\llbracket x \in SComplex; e \in Infinitesimal \rrbracket \implies stc(e + x) = x$
 $\langle proof \rangle$

lemma *HFinite-stc-Infinitesimal-add*:
 $x \in HFinite \implies \exists e \in Infinitesimal. x = stc(x) + e$

$\langle \text{proof} \rangle$

lemma *stc-add*:

$\llbracket x \in \text{HFinite}; y \in \text{HFinite} \rrbracket \implies \text{stc } (x + y) = \text{stc}(x) + \text{stc}(y)$
 $\langle \text{proof} \rangle$

lemma *stc-zero*: $\text{stc } 0 = 0$

$\langle \text{proof} \rangle$

lemma *stc-one*: $\text{stc } 1 = 1$

$\langle \text{proof} \rangle$

lemma *stc-minus*: $y \in \text{HFinite} \implies \text{stc}(-y) = -\text{stc}(y)$

$\langle \text{proof} \rangle$

lemma *stc-diff*:

$\llbracket x \in \text{HFinite}; y \in \text{HFinite} \rrbracket \implies \text{stc } (x - y) = \text{stc}(x) - \text{stc}(y)$
 $\langle \text{proof} \rangle$

lemma *stc-mult*:

$\llbracket x \in \text{HFinite}; y \in \text{HFinite} \rrbracket$
 $\implies \text{stc } (x * y) = \text{stc}(x) * \text{stc}(y)$
 $\langle \text{proof} \rangle$

lemma *stc-Infinitesimal*: $x \in \text{Infinitesimal} \implies \text{stc } x = 0$

$\langle \text{proof} \rangle$

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

$\langle \text{proof} \rangle$

lemma *stc-inverse*:

$\llbracket x \in \text{HFinite}; \text{stc } x \neq 0 \rrbracket \implies \text{stc}(\text{inverse } x) = \text{inverse } (\text{stc } x)$
 $\langle \text{proof} \rangle$

lemma *stc-divide* [simp]:

$\llbracket x \in \text{HFinite}; y \in \text{HFinite}; \text{stc } y \neq 0 \rrbracket$
 $\implies \text{stc}(x/y) = (\text{stc } x) / (\text{stc } y)$
 $\langle \text{proof} \rangle$

lemma *stc-idempotent* [simp]: $x \in \text{HFinite} \implies \text{stc}(\text{stc}(x)) = \text{stc}(x)$

$\langle \text{proof} \rangle$

lemma *HFinite-HFinite-hcomplex-of-hypreal*:

$z \in \text{HFinite} \implies \text{hcomplex-of-hypreal } z \in \text{HFinite}$
 $\langle \text{proof} \rangle$

lemma *SComplex-SReal-hcomplex-of-hypreal*:

$x \in \mathbb{R} \implies \text{hcomplex-of-hypreal } x \in \text{SComplex}$
 $\langle \text{proof} \rangle$

lemma *stc-hcomplex-of-hypreal*:

$z \in HFinite \implies stc(hcomplex-of-hypreal\ z) = hcomplex-of-hypreal\ (st\ z)$
 $\langle proof \rangle$

lemma *hmod-stc-eq*:

assumes $x \in HFinite$
shows $hmod(stc\ x) = st(hmod\ x)$
 $\langle proof \rangle$

lemma *Infinitesimal-hcnj-iff [simp]*:

$(hcnj\ z \in Infinitesimal) \longleftrightarrow (z \in Infinitesimal)$
 $\langle proof \rangle$

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*: $*s* X \cap SComplex = hcomplex-of-complex\ 'X$
 $\langle proof \rangle$

lemma *lemma-not-hcomplexA*: $x \notin hcomplex-of-complex\ 'A \implies \forall y \in A. x \neq hcomplex-of-complex\ y$
 $\langle proof \rangle$

16.2 Theorems about Nonstandard Extensions of Functions

lemma *starfunC-hcpow*: $\bigwedge Z. (*f* (\lambda z. z \hat{\ } n))\ Z = Z\ pow\ hypnat-of-nat\ n$
 $\langle proof \rangle$

lemma *starfunCR-cmod*: $*f*\ cmod = hmod$
 $\langle proof \rangle$

16.3 Internal Functions - Some Redundancy With *f* Now

lemma *starfun-Re*: $(*f* (\lambda x. Re\ (f\ x))) = (\lambda x. hRe\ ((*f*\ f)\ x))$
 $\langle proof \rangle$

lemma *starfun-Im*: $(*f* (\lambda x. Im\ (f\ x))) = (\lambda x. hIm\ ((*f*\ f)\ x))$
 $\langle proof \rangle$

lemma *starfunC-eq-Re-Im-iff*:

$(\ast f \ast f) x = z \longleftrightarrow (\ast f \ast (\lambda x. \text{Re } (f x))) x = \text{hRe } z \wedge (\ast f \ast (\lambda x. \text{Im } (f x))) x = \text{hIm } z$
 $\langle \text{proof} \rangle$

lemma *starfunC-approx-Re-Im-iff*:

$(\ast f \ast f) x \approx z \longleftrightarrow (\ast f \ast (\lambda x. \text{Re } (f x))) x \approx \text{hRe } z \wedge (\ast f \ast (\lambda x. \text{Im } (f x))) x \approx \text{hIm } z$
 $\langle \text{proof} \rangle$

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]: $x \neq 0 \implies x \ast (\text{inverse } x)^2 = \text{inverse } x$

for $x :: \text{complex}$

$\langle \text{proof} \rangle$

Changing the quantified variable. Install earlier?

lemma *all-shift*: $(\forall x :: 'a :: \text{comm-ring-1}. P x) \longleftrightarrow (\forall x. P (x - a))$

$\langle \text{proof} \rangle$

17.1 Limit of Complex to Complex Function

lemma *NSLIM-Re*: $f -a \rightarrow_{NS} L \implies (\lambda x. \text{Re } (f x)) -a \rightarrow_{NS} \text{Re } L$

$\langle \text{proof} \rangle$

lemma *NSLIM-Im*: $f -a \rightarrow_{NS} L \implies (\lambda x. \text{Im } (f x)) -a \rightarrow_{NS} \text{Im } L$

$\langle \text{proof} \rangle$

lemma *LIM-Re*: $f -a \rightarrow L \implies (\lambda x. \text{Re } (f x)) -a \rightarrow \text{Re } L$

for $f :: 'a :: \text{real-normed-vector} \Rightarrow \text{complex}$

$\langle \text{proof} \rangle$

lemma *LIM-Im*: $f -a \rightarrow L \implies (\lambda x. \text{Im } (f x)) -a \rightarrow \text{Im } L$

for $f :: 'a :: \text{real-normed-vector} \Rightarrow \text{complex}$

$\langle \text{proof} \rangle$

lemma *LIM-cn timer*: $f \rightarrow_a L \implies (\lambda x. \text{cn timer } (f x)) \rightarrow_a \text{cn timer } L$
for $f :: 'a::\text{real-normed-vector} \Rightarrow \text{complex}$
 $\langle \text{proof} \rangle$

lemma *LIM-cn timer-iff*: $((\lambda x. \text{cn timer } (f x)) \rightarrow_a \text{cn timer } L) \longleftrightarrow f \rightarrow_a L$
for $f :: 'a::\text{real-normed-vector} \Rightarrow \text{complex}$
 $\langle \text{proof} \rangle$

lemma *starfun-norm*: $(\text{starfun } (\lambda x. \text{norm } (f x))) = (\lambda x. \text{hn timer } ((\text{starfun } f) x))$
 $\langle \text{proof} \rangle$

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

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

Another equivalence result.

lemma *NSCLIM-NSCRLIM-iff*: $f \rightarrow_{NS} L \longleftrightarrow (\lambda y. \text{cmod } (f y - L)) \rightarrow_{NS} 0$
 $\langle \text{proof} \rangle$

Much, much easier standard proof.

lemma *CLIM-CRLIM-iff*: $f \rightarrow L \longleftrightarrow (\lambda y. \text{cmod } (f y - L)) \rightarrow 0$
for $f :: 'a::\text{real-normed-vector} \Rightarrow \text{complex}$
 $\langle \text{proof} \rangle$

So this is nicer nonstandard proof.

lemma *NSCLIM-NSCRLIM-iff2*: $f \rightarrow_{NS} L \longleftrightarrow (\lambda y. \text{cmod } (f y - L)) \rightarrow_{NS} 0$
 $\langle \text{proof} \rangle$

lemma *NSLIM-NSCRLIM-Re-Im-iff*:
 $f \rightarrow_{NS} L \longleftrightarrow (\lambda x. \text{Re } (f x)) \rightarrow_{NS} \text{Re } L \wedge (\lambda x. \text{Im } (f x)) \rightarrow_{NS} \text{Im } L$
 $\langle \text{proof} \rangle$

lemma *LIM-CRLIM-Re-Im-iff*: $f \rightarrow L \longleftrightarrow (\lambda x. \text{Re } (f x)) \rightarrow \text{Re } L \wedge (\lambda x. \text{Im } (f x)) \rightarrow \text{Im } L$
for $f :: 'a::\text{real-normed-vector} \Rightarrow \text{complex}$
 $\langle \text{proof} \rangle$

17.2 Continuity

lemma *NSLIM-isContc-iff*: $f \rightarrow_{NS} f a \longleftrightarrow (\lambda h. f (a + h)) \rightarrow_{NS} f a$
 $\langle \text{proof} \rangle$

17.3 Functions from Complex to Reals

lemma *isNSContCR-cmod [simp]*: $\text{isNSCont cmod } a$

$\langle \text{proof} \rangle$

lemma *isContCR-cmod* [simp]: *isCont cmod a*
 $\langle \text{proof} \rangle$

lemma *isCont-Re*: *isCont f a \implies isCont ($\lambda x. \text{Re } (f x)$) a*
for *f* :: 'a::real-normed-vector \Rightarrow complex
 $\langle \text{proof} \rangle$

lemma *isCont-Im*: *isCont f a \implies isCont ($\lambda x. \text{Im } (f x)$) a*
for *f* :: 'a::real-normed-vector \Rightarrow complex
 $\langle \text{proof} \rangle$

17.4 Differentiation of Natural Number Powers

lemma *CDERIV-pow* [simp]: *DERIV ($\lambda x. x \wedge n$) x :> complex-of-real (real n) * ($x \wedge (n - \text{Suc } 0)$)*
 $\langle \text{proof} \rangle$

Nonstandard version.

lemma *NSCDERIV-pow*: *NSDERIV ($\lambda x. x \wedge n$) x :> complex-of-real (real n) * ($x \wedge (n - 1)$)*
 $\langle \text{proof} \rangle$

Can't relax the premise $x \neq 0$: it isn't continuous at zero.

lemma *NSCDERIV-inverse*: *$x \neq 0 \implies \text{NSDERIV } (\lambda x. \text{inverse } x) x :> -(\text{inverse } x)^2$*
for *x* :: complex
 $\langle \text{proof} \rangle$

lemma *CDERIV-inverse*: *$x \neq 0 \implies \text{DERIV } (\lambda x. \text{inverse } x) x :> -(\text{inverse } x)^2$*
for *x* :: complex
 $\langle \text{proof} \rangle$

17.5 Derivative of Reciprocals (Function *inverse*)

lemma *CDERIV-inverse-fun*:
*DERIV f x :> d $\implies f x \neq 0 \implies \text{DERIV } (\lambda x. \text{inverse } (f x)) x :> -(d * \text{inverse } ((f x)^2))$*
for *x* :: complex
 $\langle \text{proof} \rangle$

lemma *NSCDERIV-inverse-fun*:
*NSDERIV f x :> d $\implies f x \neq 0 \implies \text{NSDERIV } (\lambda x. \text{inverse } (f x)) x :> -(d * \text{inverse } ((f x)^2))$*
for *x* :: complex
 $\langle \text{proof} \rangle$

17.6 Derivative of Quotient

lemma *CDERIV-quotient*:

$DERIV f x :> d \implies DERIV g x :> e \implies g(x) \neq 0 \implies$
 $DERIV (\lambda y. f y / g y) x :> (d * g x - (e * f x)) / (g x)^2$
for $x :: complex$
 $\langle proof \rangle$

lemma *NSCDERIV-quotient*:

$NSDERIV f x :> d \implies NSDERIV g x :> e \implies g x \neq (0 :: complex) \implies$
 $NSDERIV (\lambda y. f y / g y) x :> (d * g x - (e * f x)) / (g x)^2$
 $\langle proof \rangle$

17.7 Caratheodory Formulation of Derivative at a Point: Standard Proof

lemma *CARAT-CDERIVD*:

$(\forall z. f z - f x = g z * (z - x)) \wedge isNSCont g x \wedge g x = l \implies NSDERIV f x :> l$
 $\langle proof \rangle$

end

18 Logarithms: Non-Standard Version

theory *HLog*

imports *HTranscendental*

begin

definition *powhr* :: $hypreal \Rightarrow hypreal \Rightarrow hypreal$ (**infixr** $\langle powhr \rangle$ 80)
where $[transfer-unfold]: x \text{ powhr } a = starfun2 (powr) x a$

definition *hlog* :: $hypreal \Rightarrow hypreal \Rightarrow hypreal$
where $[transfer-unfold]: hlog a x = starfun2 log a x$

lemma *powhr*: $(star-n X) \text{ powhr } (star-n Y) = star-n (\lambda n. (X n) \text{ powr } (Y n))$
 $\langle proof \rangle$

lemma *powhr-one-eq-one* $[simp]: \bigwedge a. 1 \text{ powhr } a = 1$
 $\langle proof \rangle$

lemma *powhr-mult*: $\bigwedge a x y. 0 < x \implies 0 < y \implies (x * y) \text{ powhr } a = (x \text{ powhr } a) * (y \text{ powhr } a)$
 $\langle proof \rangle$

lemma *powhr-gt-zero* $[simp]: \bigwedge a x. 0 < x \text{ powhr } a \longleftrightarrow x \neq 0$
 $\langle proof \rangle$

lemma *powhr-not-zero* $[simp]: \bigwedge a x. x \text{ powhr } a \neq 0 \longleftrightarrow x \neq 0$
 $\langle proof \rangle$

lemma powhr-divide: $\bigwedge a \ x \ y. \ 0 \leq x \implies 0 \leq y \implies (x / y) \text{ powhr } a = (x \text{ powhr } a) / (y \text{ powhr } a)$
 $\langle \text{proof} \rangle$

lemma powhr-add: $\bigwedge a \ b \ x. \ x \text{ powhr } (a + b) = (x \text{ powhr } a) * (x \text{ powhr } b)$
 $\langle \text{proof} \rangle$

lemma powhr-powhr: $\bigwedge a \ b \ x. \ (x \text{ powhr } a) \text{ powhr } b = x \text{ powhr } (a * b)$
 $\langle \text{proof} \rangle$

lemma powhr-powhr-swap: $\bigwedge a \ b \ x. \ (x \text{ powhr } a) \text{ powhr } b = (x \text{ powhr } b) \text{ powhr } a$
 $\langle \text{proof} \rangle$

lemma powhr-minus: $\bigwedge a \ x. \ x \text{ powhr } (- a) = \text{inverse } (x \text{ powhr } a)$
 $\langle \text{proof} \rangle$

lemma powhr-minus-divide: $x \text{ powhr } (- a) = 1 / (x \text{ powhr } a)$
 $\langle \text{proof} \rangle$

lemma powhr-less-mono: $\bigwedge a \ b \ x. \ a < b \implies 1 < x \implies x \text{ powhr } a < x \text{ powhr } b$
 $\langle \text{proof} \rangle$

lemma powhr-less-cancel: $\bigwedge a \ b \ x. \ x \text{ powhr } a < x \text{ powhr } b \implies 1 < x \implies a < b$
 $\langle \text{proof} \rangle$

lemma powhr-less-cancel-iff [simp]: $1 < x \implies x \text{ powhr } a < x \text{ powhr } b \longleftrightarrow a < b$
 $\langle \text{proof} \rangle$

lemma powhr-le-cancel-iff [simp]: $1 < x \implies x \text{ powhr } a \leq x \text{ powhr } b \longleftrightarrow a \leq b$
 $\langle \text{proof} \rangle$

lemma hlog: $\text{hlog } (\text{star-}n \ X) \ (\text{star-}n \ Y) = \text{star-}n \ (\lambda n. \log \ (X \ n) \ (Y \ n))$
 $\langle \text{proof} \rangle$

lemma hlog-starfun-ln: $\bigwedge x. \ (*f* \ \ln) \ x = \text{hlog } ((*f* \ \exp) \ 1) \ x$
 $\langle \text{proof} \rangle$

lemma powhr-hlog-cancel [simp]: $\bigwedge a \ x. \ 0 < a \implies a \neq 1 \implies 0 < x \implies a \text{ powhr } (\text{hlog } a \ x) = x$
 $\langle \text{proof} \rangle$

lemma hlog-powhr-cancel [simp]: $\bigwedge a \ y. \ 0 < a \implies a \neq 1 \implies \text{hlog } a \ (a \text{ powhr } y) = y$
 $\langle \text{proof} \rangle$

lemma hlog-mult:
 $\bigwedge a \ x \ y. \ \text{hlog } a \ (x * y) = (\text{if } x \neq 0 \ \wedge \ y \neq 0 \text{ then } \text{hlog } a \ x + \text{hlog } a \ y \text{ else } 0)$
 $\langle \text{proof} \rangle$

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*:
 $x \in \text{HFinite} \implies 0 < x \implies a \in \text{HFinite} - \text{Infinitesimal} \implies 0 < a \implies x \text{ powhr } a \in \text{HFinite}$
 ⟨proof⟩

lemma *hlog-hrabs-HInfinite-Infinitesimal*:
 $x \in \text{HFinite} - \text{Infinitesimal} \implies a \in \text{HFinite} \implies 0 < a \implies \text{hlog } a \ |x| \in \text{Infinitesimal}$
 ⟨proof⟩

lemma *hlog-HInfinite-as-starfun*: $a \in \text{HFinite} \implies 0 < a \implies \text{hlog } a \ x = (*f* \ \text{ln}) \ x / (*f* \ \text{ln}) \ a$
 ⟨proof⟩

lemma *hlog-one [simp]*: $\bigwedge a. \text{hlog } a \ 1 = 0$
 ⟨proof⟩

lemma *hlog-eq-one [simp]*: $\bigwedge a. 0 < a \implies a \neq 1 \implies \text{hlog } a \ a = 1$
 ⟨proof⟩

lemma *hlog-inverse*: $\bigwedge a x. \text{hlog } a \ (\text{inverse } x) = - \text{hlog } a \ x$
 ⟨proof⟩

lemma *hlog-divide*: $\text{hlog } a \ (x / y) = (\text{if } x \neq 0 \wedge y \neq 0 \text{ then } \text{hlog } a \ x - \text{hlog } a \ y \text{ else } 0)$
 ⟨proof⟩

lemma *hlog-less-cancel-iff [simp]*:
 $\bigwedge a x y. 1 < a \implies 0 < x \implies 0 < y \implies \text{hlog } a \ x < \text{hlog } a \ y \longleftrightarrow x < y$
 ⟨proof⟩

lemma *hlog-le-cancel-iff [simp]*: $1 < a \implies 0 < x \implies 0 < y \implies \text{hlog } a \ x \leq \text{hlog } a \ y \longleftrightarrow x \leq y$
 ⟨proof⟩

end

theory *Hyperreal*
imports *HLog*
begin

end
theory *Hypercomplex*
imports *CLim Hyperreal*
begin

end

theory *Nonstandard-Analysis*
imports *Hypercomplex*
begin

end