Actuarial Mathematics

Yosuke Ito

January 25, 2022

Abstract

Actuarial Mathematics is a theory in applied mathematics, which is mainly used for determining the prices of insurance products and evaluating the liability of a company associating with insurance contracts. It is related to calculus, probability theory and financial theory, etc.

In this entry, I formalize the very basic part of Actuarial Mathematics in Isabelle/HOL. The first formalization is about the theory of interest which deals with interest rates, present value factors, an annuity certain, etc.

I have already formalized the basic part of Actuarial Mathematics in Coq (https://github.com/Yosuke-Ito-345/Actuary). This entry is currently the partial translation and a little generalization of the Coq formalization. The further translation in Isabelle/HOL is now proceeding.

Contents

theory *Preliminaries*
imports *HOL−Analysis.Analysis*
begin

notation *powr* (**infixr** .^ *80*)

1 Preliminary Definitions and Lemmas

lemma *seq-part-multiple*: **fixes** $m\ n :: nat$ **assumes** $m \neq 0$ **defines** $A \equiv \lambda i{::}nat.\ \{i{*}m\ ..{<}\ (i{+}1){*}m\}$
shows $\forall i\ j.\ i \neq j \longrightarrow A\ i \cap A\ j = \{\}$ **and** $(\bigcup i{<}n.\ A\ i) = \{..{<}\ n{*}m\}$
⟨*proof*⟩

lemma(**in** *field*) *divide-mult-cancel*[*simp*]: **fixes** a b **assumes** $b \neq 0$
shows a / $b * b = a$
⟨*proof*⟩

lemma *inverse-powr*: $(1/a).\hat{}b = a.\hat{}{-b}$ **if** $a > 0$ **for** a b :: *real*
⟨*proof*⟩

lemma *powr-eq-one-iff-gen*[*simp*]: $a.\hat{}x = 1 \longleftrightarrow x = 0$ **if** $a > 0$ $a \neq 1$ **for** a x :: *real*
⟨*proof*⟩

lemma *powr-less-cancel2*: $0 < a \Longrightarrow 0 < x \Longrightarrow 0 < y \Longrightarrow x.\hat{}a < y.\hat{}a \Longrightarrow x < y$
for a x y ::*real*
⟨*proof*⟩

lemma *geometric-increasing-sum-aux*: $(1-r)\hat{}2 * (\sum k<n. (k+1)*r\hat{}k) = 1 - (n+1)*r\hat{}n + n*r\hat{}(n+1)$
for n::*nat* **and** r::*real*
⟨*proof*⟩

lemma *geometric-increasing-sum*: $(\sum k<n. (k+1)*r\hat{}k) = (1 - (n+1)*r\hat{}n + n*r\hat{}(n+1)) / (1-r)\hat{}2$
if $r \neq 1$ **for** n::*nat* **and** r::*real*
⟨*proof*⟩

lemma *Reals-UNIV*[*simp*]: $\mathbb{R} = \{x$::*real*. *True*$\}$
⟨*proof*⟩

lemma *DERIV-fun-powr2*:
fixes a::*real*
assumes *a-pos*: $a > 0$
and f: *DERIV* f x :> r
shows *DERIV* $(\lambda x. a.\hat{}(f\ x))$ x :> $a.\hat{}(f\ x) * r * ln\ a$
⟨*proof*⟩

lemma *has-real-derivative-powr2*:
assumes *a-pos*: $a > 0$
shows $((\lambda x. a.\hat{}x)$ *has-real-derivative* $a.\hat{}x * ln\ a)$ (*at* x)
⟨*proof*⟩

lemma *has-integral-powr2-from-0*:
fixes a c :: *real*
assumes *a-pos*: $a > 0$ **and** *a-neq-1*: $a \neq 1$ **and** *c-nneg*: $c \geq 0$
shows $((\lambda x. a.\hat{}x)$ *has-integral* $((a.\hat{}c - 1) / (ln\ a)))$ $\{0..c\}$
⟨*proof*⟩

lemma *integrable-on-powr2-from-0*:
fixes a c :: *real*

assumes *a-pos*: $a > 0$ **and** *a-neq-1*: $a \neq 1$ **and** *c-nneg*: $c \geq 0$
shows $(\lambda x.\ a.\hat{}x)$ *integrable-on* $\{0..c\}$
⟨*proof*⟩

lemma *integrable-on-powr2-from-0-general*:
fixes *a c* :: *real*
assumes *a-pos*: $a > 0$ **and** *c-nneg*: $c \geq 0$
shows $(\lambda x.\ a.\hat{}x)$ *integrable-on* $\{0..c\}$
⟨*proof*⟩

lemma *has-integral-null-interval*: **fixes** *a b* :: *real* **and** *f*::*real* ⇒ *real* **assumes** $a \geq b$
shows (*f has-integral 0*) $\{a..b\}$
⟨*proof*⟩

lemma *has-integral-interval-reverse*: **fixes** *f* :: *real* ⇒ *real* **and** *a b* :: *real*
assumes $a \leq b$
and *continuous-on* $\{a..b\}$ *f*
shows $((\lambda x.\ f\ (a+b-x))$ *has-integral* (*integral* $\{a..b\}$ *f*)) $\{a..b\}$
⟨*proof*⟩

end
theory *Interest*
imports *Preliminaries*
begin

2 List of Actuarial Notations (Global Scope)

definition *i-nom* :: *real* ⇒ *nat* ⇒ *real* ($*i*[-]^{-} [*0*,*0*] *200*)
where $*i*[*i*]^{*m*} ≡ $m * ((1+i).\hat{}(1/m) - 1)$ — nominal interest rate
definition *i-force* :: *real* ⇒ *real* ($*δ*[-] [*0*] *200*)
where $*δ*[*i*] ≡ *ln* (*1*+*i*) — force of interest
definition *d-nom* :: *real* ⇒ *nat* ⇒ *real* ($*d*[-]^{-} [*0*,*0*] *200*)
where $*d*[*i*]^{*m*} ≡ $*i*[*i*]^{*m*} / (*1* + $*i*[*i*]^{*m*}/*m*) — discount rate
abbreviation *d-nom-yr* :: *real* ⇒ *real* ($*d*[-] [*0*] *200*)
where $*d*[*i*] ≡ $*d*[*i*]^{*1*} — Post-fix "yr" stands for "year".
definition *v-pres* :: *real* ⇒ *real* ($*v*[-] [*0*] *200*)
where $*v*[*i*] ≡ *1* / (*1*+*i*) — present value factor
definition *ann* :: *real* ⇒ *nat* ⇒ *nat* ⇒ *real* ($*a*[-]^{-}′-- [*0*,*0*,*101*] *200*)
where $*a*[*i*]^{*m*}-*n* ≡ ∑ *k*<*n*∗*m*. $*v*[*i*].^((*k*+*1*::*nat*)/*m*) / *m*
— present value of an immediate annuity
abbreviation *ann-yr* :: *real* ⇒ *nat* ⇒ *real* ($*a*[-]′-- [*0*,*101*] *200*)
where $*a*[*i*]-*n* ≡ $*a*[*i*]^{*1*}-*n*
definition *acc* :: *real* ⇒ *nat* ⇒ *nat* ⇒ *real* ($*s*[-]^{-}′-- [*0*,*0*,*101*] *200*)
where $*s*[*i*]^{*m*}-*n* ≡ ∑ *k*<*n*∗*m*. (*1*+*i*).^((*k*::*nat*)/*m*) / *m*
— future value of an immediate annuity
— The name "acc" stands for "accumulation".
abbreviation *acc-yr* :: *real* ⇒ *nat* ⇒ *real* ($*s*[-]′-- [*0*] *200*)
where $*s*[*i*]-*n* ≡ $*s*[*i*]^{*1*}-*n*

```
definition ann-due :: real ⇒ nat ⇒ nat ⇒ real ($a''''[-]^{-}'-- [0,0,101] 200)
  where $a''[i]^{m}-n ≡ ∑ k<n*m. $v[i].^((k::nat)/m) / m
    — present value of an annuity-due
abbreviation ann-due-yr :: real ⇒ nat ⇒ real ($a''''[-]'-- [0,101] 200)
  where $a''[i]-n ≡ $a''[i]^{1}-n
definition acc-due :: real ⇒ nat ⇒ nat ⇒ real ($s''''[-]^{-}'-- [0,0,101] 200)
  where $s''[i]^{m}-n ≡ ∑ k<n*m. (1+i).^((k+1::nat)/m) / m
    — future value of an annuity-due
abbreviation acc-due-yr :: real ⇒ nat ⇒ real ($s''''[-]'-- [0,101] 200)
  where $s''[i]-n ≡ $s''[i]^{1}-n
definition ann-cont :: real ⇒ real ⇒ real ($a''[-]'-- [0,101] 200)
  where $a'[i]-n ≡ integral {0..n} (λt::real. $v[i].^t)
    — present value of a continuous annuity
definition acc-cont :: real ⇒ real ⇒ real ($s''[-]'-- [0,101] 200)
  where $s'[i]-n ≡ integral {0..n} (λt::real. (1+i).^t)
    — future value of a continuous annuity
definition perp :: real ⇒ nat ⇒ real ($a[-]^{-}'-∞ [0,0] 200)
  where $a[i]^{m}-∞ ≡ 1 / $i[i]^{m}
    — present value of a perpetual annuity
abbreviation perp-yr :: real ⇒ real ($a[-]'-∞ [0] 200)
  where $a[i]-∞ ≡ $a[i]^{1}-∞
definition perp-due :: real ⇒ nat ⇒ real ($a''''[-]^{-}'-∞ [0,0] 200)
  where $a''[i]^{m}-∞ ≡ 1 / $d[i]^{m}
    — present value of a perpetual annuity-due
abbreviation perp-due-yr :: real ⇒ real ($a''''[-]'-∞ [0] 200)
  where $a''[i]-∞ ≡ $a''[i]^{1}-∞
definition ann-incr :: nat ⇒ real ⇒ nat ⇒ nat ⇒ real
  ($'(I^{-}a')[-]^{-}'-- [0,0,0,101] 200)
 where $(I^{l}a)[i]^{m}-n ≡ ∑ k<n*m. $v[i].^((k+1::nat)/m) * ⌈l*(k+1::nat)/m⌉
/ (l*m)
    — present value of an increasing annuity
    — This is my original definition.
    — Here, "l" represents the number of increments per unit time.
abbreviation ann-incr-lvl :: real ⇒ nat ⇒ nat ⇒ real
  ($'(Ia')[-]^{-}'-- [0,0,101] 200)
  where $(Ia)[i]^{m}-n ≡ $(I^{1}a)[i]^{m}-n
    — The post-fix "lvl" stands for "level".
abbreviation ann-incr-yr :: real ⇒ nat ⇒ real ($'(Ia')[-]'-- [0,101] 200)
  where $(Ia)[i]-n ≡ $(Ia)[i]^{1}-n
definition acc-incr :: nat ⇒ real ⇒ nat ⇒ nat ⇒ real
  ($'(I^{-}s')[-]^{-}'-- [0,0,0,101] 200)
 where $(I^{l}s)[i]^{m}-n ≡ ∑ k<n*m. (1+i).^(n−(k+1::nat)/m) * ⌈l*(k+1::nat)/m⌉
/ (l*m)
    — future value of an increasing annuity
abbreviation acc-incr-lvl :: real ⇒ nat ⇒ nat ⇒ real
  ($'(Is')[-]^{-}'-- [0,0,101] 200)
  where $(Is)[i]^{m}-n ≡ $(I^{1}s)[i]^{m}-n
abbreviation acc-incr-yr :: real ⇒ nat ⇒ real ($'(Is')[-]'-- [0,101] 200)
  where $(Is)[i]-n ≡ $(Is)[i]^{1}-n
```

definition *ann-due-incr* :: *nat* ⇒ *real* ⇒ *nat* ⇒ *nat* ⇒ *real*
($'(I^{-}a'''')[-]^{-}'-- [0,0,0,101] 200)
where $(I^{l}a'')[i]^{m}-n ≡ ∑ k<n*m. $v[i].^((k::nat)/m) * ⌈l*(k+1::nat)/m⌉ / (l*m)
abbreviation *ann-due-incr-lvl* :: *real* ⇒ *nat* ⇒ *nat* ⇒ *real*
($'(Ia'''')[-]^{-}'-- [0,0,101] 200)
where $(Ia'')[i]^{m}-n ≡ $(I^{1}a'')[i]^{m}-n
abbreviation *ann-due-incr-yr* :: *real* ⇒ *nat* ⇒ *real* ($'(Ia'''')[-]'-- [0,101] 200)
where $(Ia'')[i]-n ≡ $(Ia'')[i]^{1}-n
definition *acc-due-incr* :: *nat* ⇒ *real* ⇒ *nat* ⇒ *nat* ⇒ *real*
($'(I^{-}s'''')[-]^{-}'-- [0,0,0,101] 200)
where $(I^{l}s'')[i]^{m}-n ≡ ∑ k<n*m. (1+i).^(n−(k::nat)/m) * ⌈l*(k+1::nat)/m⌉ / (l*m)
abbreviation *acc-due-incr-lvl* :: *real* ⇒ *nat* ⇒ *nat* ⇒ *real*
($'(Is'''')[-]^{-}'-- [0,0,101] 200)
where $(Is'')[i]^{m}-n ≡ $(I^{1}s'')[i]^{m}-n
abbreviation *acc-due-incr-yr* :: *real* ⇒ *nat* ⇒ *real* ($'(Is'''')[-]'-- [0,101] 200)
where $(Is'')[i]-n ≡ $(Is'')[i]^{1}-n
definition *perp-incr* :: *nat* ⇒ *real* ⇒ *nat* ⇒ *real* ($'(I^{-}a')[-]^{-}'-∞ [0,0,0] 200)
where $(I^{l}a)[i]^{m}-∞ ≡ *lim* (λn. $(I^{l}a)[i]^{m}-n)
abbreviation *perp-incr-lvl* :: *real* ⇒ *nat* ⇒ *real* ($'(Ia')[-]^{-}'-∞ [0,0] 200)
where $(Ia)[i]^{m}-∞ ≡ $(I^{1}a)[i]^{m}-∞
abbreviation *perp-incr-yr* :: *real* ⇒ *real* ($'(Ia')[-]'-∞ [0] 200)
where $(Ia)[i]-∞ ≡ $(Ia)[i]^{1}-∞
definition *perp-due-incr* :: *nat* ⇒ *real* ⇒ *nat* ⇒ *real* ($'(I^{-}a'''')[-]^{-}'-∞ [0,0,0] 200)
where $(I^{l}a'')[i]^{m}-∞ ≡ *lim* (λn. $(I^{l}a'')[i]^{m}-n)
abbreviation *perp-due-incr-lvl* :: *real* ⇒ *nat* ⇒ *real* ($'(Ia'''')[-]^{-}'-∞ [0,0] 200)
where $(Ia'')[i]^{m}-∞ ≡ $(I^{1}a'')[i]^{m}-∞
abbreviation *perp-due-incr-yr* :: *real* ⇒ *real* ($'(Ia'''')[-]'-∞ [0] 200)
where $(Ia'')[i]-∞ ≡ $(Ia'')[i]^{1}-∞

3 Theory of Interest

locale *interest* =
fixes *i* :: *real* — i stands for an interest rate.
assumes *v-futr-pos*: *1* + *i* > *0* — Assume that the future value is positive.

context *interest*
begin

abbreviation *i-nom'* :: *nat* ⇒ *real* ($i^{-} [0] 200)
where $i^{m} ≡ $i[i]^{m}
abbreviation *i-force'* :: *real* ($δ)
where $δ ≡ $δ[i]
abbreviation *d-nom'* :: *nat* ⇒ *real* ($d^{-} [0] 200)
where $d^{m} ≡ $d[i]^{m}

abbreviation *d-nom-yr'* :: *real* ($*d*)
  where $*d* ≡ $*d*[*i*]
abbreviation *v-pres'* :: *real* ($*v*)
  where $*v* ≡ $*v*[*i*]
abbreviation *ann'* :: *nat* ⇒ *nat* ⇒ *real* ($*a*^{-}'-- [*0*,*101*] *200*)
  where $*a*^{*m*}-*n* ≡ $*a*[*i*]^{*m*}-*n*
abbreviation *ann-yr'* :: *nat* ⇒ *real* ($*a*'-- [*101*] *200*)
  where $*a*-*n* ≡ $*a*[*i*]-*n*
abbreviation *acc'* :: *nat* ⇒ *nat* ⇒ *real* ($*s*^{-}'-- [*0*,*101*] *200*)
  where $*s*^{*m*}-*n* ≡ $*s*[*i*]^{*m*}-*n*
abbreviation *acc-yr'* :: *nat* ⇒ *real* ($*s*'-- [*101*] *200*)
  where $*s*-*n* ≡ $*s*[*i*]-*n*
abbreviation *ann-due'* :: *nat* ⇒ *nat* ⇒ *real* ($*a*''''^{-}'-- [*0*,*101*] *200*)
  where $*a*''^{*m*}-*n* ≡ $*a*''[*i*]^{*m*}-*n*
abbreviation *ann-due-yr'* :: *nat* ⇒ *real* ($*a*'''''-- [*101*] *200*)
  where $*a*''-*n* ≡ $*a*''[*i*]-*n*
abbreviation *acc-due'* :: *nat* ⇒ *nat* ⇒ *real* ($*s*''''^{-}'-- [*0*,*101*] *200*)
  where $*s*''^{*m*}-*n* ≡ $*s*''[*i*]^{*m*}-*n*
abbreviation *acc-due-yr'* :: *nat* ⇒ *real* ($*s*'''''-- [*101*] *200*)
  where $*s*''-*n* ≡ $*s*''[*i*]-*n*
abbreviation *ann-cont'* :: *real* ⇒ *real* ($*a*'''-- [*101*] *200*)
  where $*a*'-*n* ≡ $*a*'[*i*]-*n*
abbreviation *acc-cont'* :: *real* ⇒ *real* ($*s*'''-- [*101*] *200*)
  where $*s*'-*n* ≡ $*s*'[*i*]-*n*
abbreviation *perp'* :: *nat* ⇒ *real* ($*a*^{-}'-∞ [*0*] *200*)
  where $*a*^{*m*}-∞ ≡ $*a*[*i*]^{*m*}-∞
abbreviation *perp-yr'* :: *real* ($*a*'-∞)
  where $*a*-∞ ≡ $*a*[*i*]-∞
abbreviation *perp-due'* :: *nat* ⇒ *real* ($*a*''''^{-}'-∞ [*0*] *200*)
  where $*a*''^{*m*}-∞ ≡ $*a*''[*i*]^{*m*}-∞
abbreviation *perp-due-yr'* :: *real* ($*a*'''''-∞)
  where $*a*''-∞ ≡ $*a*''[*i*]-∞
abbreviation *ann-incr'* :: *nat* ⇒ *nat* ⇒ *nat* ⇒ *real* ($'(*I*^{-}*a*')^{-}'-- [*0*,*0*,*101*] *200*)
  where $(*I*^{*l*}*a*)^{*m*}-*n* ≡ $(*I*^{*l*}*a*)[*i*]^{*m*}-*n*
abbreviation *ann-incr-lvl'* :: *nat* ⇒ *nat* ⇒ *real* ($'(*Ia*')^{-}'-- [*0*,*101*] *200*)
  where $(*Ia*)^{*m*}-*n* ≡ $(*Ia*)[*i*]^{*m*}-*n*
abbreviation *ann-incr-yr'* :: *nat* ⇒ *real* ($'(*Ia*')'-- [*101*] *200*)
  where $(*Ia*)-*n* ≡ $(*Ia*)[*i*]-*n*
abbreviation *acc-incr'* :: *nat* ⇒ *nat* ⇒ *nat* ⇒ *real* ($'(*I*^{-}*s*')^{-}'-- [*0*,*0*,*101*] *200*)
  where $(*I*^{*l*}*s*)^{*m*}-*n* ≡ $(*I*^{*l*}*s*)[*i*]^{*m*}-*n*
abbreviation *acc-incr-lvl'* :: *nat* ⇒ *nat* ⇒ *real* ($'(*Is*')^{-}'-- [*0*,*101*] *200*)
  where $(*Is*)^{*m*}-*n* ≡ $(*Is*)[*i*]^{*m*}-*n*
abbreviation *acc-incr-yr'* :: *nat* ⇒ *real* ($'(*Is*')'-- [*101*] *200*)
  where $(*Is*)-*n* ≡ $(*Is*)[*i*]-*n*
abbreviation *ann-due-incr'* :: *nat* ⇒ *nat* ⇒ *nat* ⇒ *real* ($'(*I*^{-}*a*'''''')^{-}'-- [*0*,*0*,*101*] *200*)
  where $(*I*^{*l*}*a*'')^{*m*}-*n* ≡ $(*I*^{*l*}*a*'')[*i*]^{*m*}-*n*

abbreviation *ann-due-incr-lvl′* :: *nat* ⇒ *nat* ⇒ *real* ($′(*Ia′′′′*)^{-}′-- [*0*,*101*] *200*)
where $(*Ia′′*)^{*m*}-*n* ≡ $(*Ia′′*)[*i*]^{*m*}-*n*
abbreviation *ann-due-incr-yr′* :: *nat* ⇒ *real* ($′(*Ia′′′′*)′-- [*101*] *200*)
where $(*Ia′′*)-*n* ≡ $(*Ia′′*)[*i*]-*n*
abbreviation *acc-due-incr′* :: *nat* ⇒ *nat* ⇒ *nat* ⇒ *real* ($′(*I*^{-}*s′′′′*)^{-}′-- [*0*,*0*,*101*] *200*)
where $(*I*^{*l*}*s′′*)^{*m*}-*n* ≡ $(*I*^{*l*}*s′′*)[*i*]^{*m*}-*n*
abbreviation *acc-due-incr-lvl′* :: *nat* ⇒ *nat* ⇒ *real* ($′(*Is′′′′*)^{-}′-- [*0*,*101*] *200*)
where $(*Is′′*)^{*m*}-*n* ≡ $(*Is′′*)[*i*]^{*m*}-*n*
abbreviation *acc-due-incr-yr′* :: *nat* ⇒ *real* ($′(*Is′′′′*)′-- [*101*] *200*)
where $(*Is′′*)-*n* ≡ $(*Is′′*)[*i*]-*n*
abbreviation *perp-incr′* :: *nat* ⇒ *nat* ⇒ *real* ($′(*I*^{-}*a′*)^{-}′-∞ [*0*,*0*] *200*)
where $(*I*^{*l*}*a*)^{*m*}-∞ ≡ $(*I*^{*l*}*a*)[*i*]^{*m*}-∞
abbreviation *perp-incr-lvl′* :: *nat* ⇒ *real* ($′(*Ia′*)^{-}′-∞ [*0*] *200*)
where $(*Ia*)^{*m*}-∞ ≡ $(*Ia*)[*i*]^{*m*}-∞
abbreviation *perp-incr-yr′* :: *real* ($′(*Ia′*)′-∞)
where $(*Ia*)-∞ ≡ $(*Ia*)[*i*]-∞
abbreviation *perp-due-incr′* :: *nat* ⇒ *nat* ⇒ *real* ($′(*I*^{-}*a′′′′*)^{-}′-∞ [*0*,*0*] *200*)
where $(*I*^{*l*}*a′′*)^{*m*}-∞ ≡ $(*I*^{*l*}*a′′*)[*i*]^{*m*}-∞
abbreviation *perp-due-incr-lvl′* :: *nat* ⇒ *real* ($′(*Ia′′′′*)^{-}′-∞ [*0*] *200*)
where $(*Ia′′*)^{*m*}-∞ ≡ $(*Ia′′*)[*i*]^{*m*}-∞
abbreviation *perp-due-incr-yr′* :: *real* ($′(*Ia′′′′*)′-∞)
where $(*Ia′′*)-∞ ≡ $(*Ia′′*)[*i*]-∞

lemma *v-futr-m-pos*: *1* + $*i*^{*m*}/*m* > *0* **if** *m* ≠ *0* **for** *m*::*nat*
⟨*proof*⟩

lemma *i-nom-1*[*simp*]: $*i*^{*1*} = *i*
⟨*proof*⟩

lemma *i-nom-eff*: (*1* + $*i*^{*m*}/*m*)^*m* = *1* + *i* **if** *m* ≠ *0* **for** *m*::*nat*
⟨*proof*⟩

lemma *i-nom-i*: *1* + $*i*^{*m*}/*m* = (*1*+*i*).^(*1*/*m*) **if** *m* ≠ *0* **for** *m*::*nat*
⟨*proof*⟩

lemma *i-nom-0-iff-i-0*: $*i*^{*m*} = *0* ⟷ *i* = *0* **if** *m* ≠ *0* **for** *m*::*nat*
⟨*proof*⟩

lemma *i-nom-pos-iff-i-pos*: $*i*^{*m*} > *0* ⟷ *i* > *0* **if** *m* ≠ *0* **for** *m*::*nat*
⟨*proof*⟩

lemma *e-delta*: *exp* $δ = *1* + *i*
⟨*proof*⟩

lemma *delta-0-iff-i-0*: $δ = *0* ⟷ *i* = *0*
⟨*proof*⟩

lemma *lim-i-nom*: $(\lambda m.\ \$i\hat{}\{m\}) \longrightarrow \$\delta$
⟨*proof*⟩

lemma *d-nom-0-iff-i-0*: $\$d\hat{}\{m\} = 0 \longleftrightarrow i = 0$ **if** $m \neq 0$ **for** $m::nat$
⟨*proof*⟩

lemma *d-nom-pos-iff-i-pos*: $\$d\hat{}\{m\} > 0 \longleftrightarrow i > 0$ **if** $m \neq 0$ **for** $m::nat$
⟨*proof*⟩

lemma *d-nom-i-nom*: $1 - \$d\hat{}\{m\}/m = 1\ /\ (1 + \$i\hat{}\{m\}/m)$ **if** $m \neq 0$ **for** $m::nat$
⟨*proof*⟩

lemma *lim-d-nom*: $(\lambda m.\ \$d\hat{}\{m\}) \longrightarrow \$\delta$
⟨*proof*⟩

lemma *v-pos*: $\$v > 0$
⟨*proof*⟩

lemma *v-1-iff-i-0*: $\$v = 1 \longleftrightarrow i = 0$
⟨*proof*⟩

lemma *v-lt-1-iff-i-pos*: $\$v < 1 \longleftrightarrow i > 0$
⟨*proof*⟩

lemma *v-i-nom*: $\$v = (1 + \$i\hat{}\{m\}/m).\hat{}{-m}$ **if** $m \neq 0$ **for** $m::nat$
⟨*proof*⟩

lemma *i-v*: $1 + i = \$v.\hat{}{-1}$
⟨*proof*⟩

lemma *i-v-powr*: $(1 + i).\hat{}a = \$v.\hat{}{-a}$ **for** $a::real$
⟨*proof*⟩

lemma *v-delta*: $ln\ \$v = -\ \$\delta$
⟨*proof*⟩

lemma *is-derive-vpow*: $DERIV\ (\lambda t.\ \$v.\hat{}t)\ t :> -\ \$\delta * \$v.\hat{}t$
⟨*proof*⟩

lemma *d-nom-v*: $\$d\hat{}\{m\} = m * (1 - \$v.\hat{}(1/m))$ **if** $m \neq 0$ **for** $m::nat$
⟨*proof*⟩

lemma *d-nom-i-nom-v*: $\$d\hat{}\{m\} = \$i\hat{}\{m\} * \$v.\hat{}(1/m)$ **if** $m \neq 0$ **for** $m::nat$
⟨*proof*⟩

lemma *a-calc*: $\$a\hat{}\{m\}\text{-}n = (1 - \$v\hat{}n)\ /\ \$i\hat{}\{m\}$ **if** $m \neq 0\ i \neq 0$ **for** $n\ m\ ::nat$
⟨*proof*⟩

lemma *a-calc-i-0*: $\$a\hat{}\{m\}\text{-}n = n$ **if** $m \neq 0\ i = 0$ **for** $n\ m\ ::\ nat$

⟨*proof*⟩

lemma *s-calc-i-0*: $s^{m}-n = n **if** m ≠ 0 i = 0 **for** n m :: nat
⟨*proof*⟩

lemma *s-a*: $s^{m}-n = (1+i)^n * $a^{m}-n **if** m ≠ 0 **for** n m :: nat
⟨*proof*⟩

lemma *s-calc*: $s^{m}-n = ((1+i)^n − 1) / $i^{m} **if** m ≠ 0 i ≠ 0 **for** n m :: nat
⟨*proof*⟩

lemma *a''-a*: $a''^{m}-n = (1+i).^(1/m) * $a^{m}-n **if** m ≠ 0 **for** m::nat
⟨*proof*⟩

lemma *a-a''*: $a^{m}-n = $v.^(1/m) * $a''^{m}-n **if** m ≠ 0 **for** m::nat
⟨*proof*⟩

lemma *a''-calc-i-0*: $a''^{m}-n = n **if** m ≠ 0 i = 0 **for** n m :: nat
⟨*proof*⟩

lemma *s''-calc-i-0*: $s''^{m}-n = n **if** m ≠ 0 i = 0 **for** n m :: nat
⟨*proof*⟩

lemma *a''-calc*: $a''^{m}-n = (1 − $v^n) / $d^{m} **if** m ≠ 0 i ≠ 0 **for** n m :: nat
⟨*proof*⟩

lemma *s''-s*: $s''^{m}-n = (1+i).^(1/m) * $s^{m}-n **if** m ≠ 0 **for** m::nat
⟨*proof*⟩

lemma *s-s''*: $s^{m}-n = $v.^(1/m) * $s''^{m}-n **if** m ≠ 0 **for** m::nat
⟨*proof*⟩

lemma *s''-calc*: $s''^{m}-n = ((1+i)^n − 1) / $d^{m} **if** m ≠ 0 i ≠ 0 **for** n m :: nat
⟨*proof*⟩

lemma *s''-a''*: $s''^{m}-n = (1+i)^n * $a''^{m}-n **if** m ≠ 0 **for** m::nat
⟨*proof*⟩

lemma *a'-calc*: $a'-n = (1 − $v.^n) / $δ **if** i ≠ 0 n ≥ 0 **for** n::real
⟨*proof*⟩

lemma *a'-calc-i-0*: $a'-n = n **if** i = 0 n ≥ 0 **for** n::real
⟨*proof*⟩

lemma *s'-calc*: $s'-n = ((1+i).^n − 1) / $δ **if** i ≠ 0 n ≥ 0 **for** n::real
⟨*proof*⟩

lemma *s'-calc-i-0*: $\$s'\text{-}n = n$ **if** $i = 0$ $n \geq 0$ **for** $n{::}real$
⟨*proof*⟩

lemma *s'-a'*: $\$s'\text{-}n = (1+i).\hat{}n * \$a'\text{-}n$ **if** $n \geq 0$ **for** $n{::}real$
⟨*proof*⟩

lemma *lim-m-a*: $(\lambda m.\ \$a\hat{}\{m\}\text{-}n) \longrightarrow \$a'\text{-}n$ **for** $n{::}nat$
⟨*proof*⟩

lemma *lim-m-a''*: $(\lambda m.\ \$a''\hat{}\{m\}\text{-}n) \longrightarrow \$a'\text{-}n$ **for** $n{::}nat$
⟨*proof*⟩

lemma *lim-m-s*: $(\lambda m.\ \$s\hat{}\{m\}\text{-}n) \longrightarrow \$s'\text{-}n$ **for** $n{::}nat$
⟨*proof*⟩

lemma *lim-m-s''*: $(\lambda m.\ \$s''\hat{}\{m\}\text{-}n) \longrightarrow \$s'\text{-}n$ **for** $n{::}nat$
⟨*proof*⟩

lemma *lim-n-a*: $(\lambda n.\ \$a\hat{}\{m\}\text{-}n) \longrightarrow \$a\hat{}\{m\}\text{-}\infty$ **if** $m \neq 0$ $i > 0$ **for** $m{::}nat$
⟨*proof*⟩

lemma *lim-n-a''*: $(\lambda n.\ \$a''\hat{}\{m\}\text{-}n) \longrightarrow \$a''\hat{}\{m\}\text{-}\infty$ **if** $m \neq 0$ $i > 0$ **for** $m{::}nat$
⟨*proof*⟩

lemma *Ilsm-Ilam*: $\$(I\hat{}\{l\}s)\hat{}\{m\}\text{-}n = (1+i)\hat{}n * \$(I\hat{}\{l\}a)\hat{}\{m\}\text{-}n$
if $l \neq 0$ $m \neq 0$ **for** $l\ n\ m :: nat$
⟨*proof*⟩

lemma *Iam-calc*: $\$(Ia)\hat{}\{m\}\text{-}n = (\sum j{<}n.\ (j+1)/m * (\sum k{=}j*m..{<}(j+1)*m.\ \$v.\hat{}((k+1)/m)))$
if $m \neq 0$ **for** $n\ m :: nat$
⟨*proof*⟩

lemma *Ism-calc*: $\$(Is)\hat{}\{m\}\text{-}n = (\sum j{<}n.\ (j+1)/m * (\sum k{=}j*m..{<}(j+1)*m.\ (1+i).\hat{}(n-(k+1)/m)))$
if $m \neq 0$ **for** $n\ m :: nat$
⟨*proof*⟩

lemma *Imam-calc-aux*: $\$(I\hat{}\{m\}a)\hat{}\{m\}\text{-}n = (\sum k{<}n*m.\ \$v.\hat{}((k+1)/m) * (k+1)$
$/\ m\hat{}2)$
if $m \neq 0$ **for** $m{::}nat$
⟨*proof*⟩

lemma *Imam-calc*:
$\$(I\hat{}\{m\}a)\hat{}\{m\}\text{-}n = (\$v.\hat{}(1/m) * (1 - (n*m+1)*\$v\hat{}n + n*m*\$v.\hat{}(n+1/m)))$
$/\ (m*(1-\$v.\hat{}(1/m)))\hat{}2$
if $i \neq 0$ $m \neq 0$ **for** $n\ m :: nat$
⟨*proof*⟩

lemma *Imam-calc-i-0*: $\$(I\hat{}\{m\}a)\hat{}\{m\}\text{-}n = (n*m+1)*n\ /\ (2*m)$ **if** $i = 0$ $m \neq 0$

for *n m* :: *nat*
⟨*proof*⟩

lemma *Imsm-calc*:
$\$(I\hat{~}\{m\}s)\hat{~}\{m\}\text{-}n = ((1+i).\hat{~}(n+1/m) - (n*m+1)*(1+i).\hat{~}(1/m) + n*m) / (m*((1+i).\hat{~}(1/m)-1))\hat{~}2$
if $i \neq 0$ $m \neq 0$ **for** *n m* :: *nat*
⟨*proof*⟩

lemma *Imsm-calc-i-0*: $\$(I\hat{~}\{m\}s)\hat{~}\{m\}\text{-}n = (n*m+1)*n / (2*m)$ **if** $i = 0$ $m \neq 0$ **for** *n m* :: *nat*
⟨*proof*⟩

lemma *Ila′′m-Ilam*: $\$(I\hat{~}\{l\}a'')\hat{~}\{m\}\text{-}n = (1+i).\hat{~}(1/m) * \$(I\hat{~}\{l\}a)\hat{~}\{m\}\text{-}n$
if $l \neq 0$ $m \neq 0$ **for** *l m n* :: *nat*
⟨*proof*⟩

lemma *Ia′′m-calc*: $\$(Ia'')\hat{~}\{m\}\text{-}n = (\sum j<n.\ (j+1)/m * (\sum k=j*m..<(j+1)*m.\ \$v.\hat{~}(k/m)))$
if $m \neq 0$ **for** *n m* :: *nat*
⟨*proof*⟩

lemma *Ima′′m-calc-aux*: $\$(I\hat{~}\{m\}a'')\hat{~}\{m\}\text{-}n = (\sum k<n*m.\ \$v.\hat{~}(k/m) * (k+1) / m\hat{~}2)$
if $m \neq 0$ **for** *m*::*nat*
⟨*proof*⟩

lemma *Ima′′m-calc*: $\$(I\hat{~}\{m\}a'')\hat{~}\{m\}\text{-}n = (1 - (n*m+1)*\$v\hat{~}n + n*m*\$v.\hat{~}(n+1/m)) / (m*(1-\$v.\hat{~}(1/m)))\hat{~}2$
if $i \neq 0$ $m \neq 0$ **for** *n m* :: *nat*
⟨*proof*⟩

lemma *Ils′′m-Ilsm*: $\$(I\hat{~}\{l\}s'')\hat{~}\{m\}\text{-}n = (1+i).\hat{~}(1/m) * \$(I\hat{~}\{l\}s)\hat{~}\{m\}\text{-}n$
if $l \neq 0$ $m \neq 0$ **for** *l m n* :: *nat*
⟨*proof*⟩

lemma *Ims′′m-calc*:
$\$(I\hat{~}\{m\}s'')\hat{~}\{m\}\text{-}n =$
$(1+i).\hat{~}(1/m) * ((1+i).\hat{~}(n+1/m) - (n*m+1)*(1+i).\hat{~}(1/m) + n*m) / (m*((1+i).\hat{~}(1/m)-1))\hat{~}2$
if $i \neq 0$ $m \neq 0$ **for** *n m* :: *nat*
⟨*proof*⟩

lemma *lim-Imam*: $(\lambda n.\ \$(I\hat{~}\{m\}a)\hat{~}\{m\}\text{-}n) \longrightarrow 1 / (\$i\hat{~}\{m\}*\$d\hat{~}\{m\})$ **if** $m \neq 0$ $i > 0$ **for** *m*::*nat*
⟨*proof*⟩

lemma *perp-incr-calc*: $\$(I\hat{~}\{m\}a)\hat{~}\{m\}\text{-}\infty = 1 / (\$i\hat{~}\{m\}*\$d\hat{~}\{m\})$ **if** $m \neq 0$ $i > 0$ **for** *m*::*nat*

⟨*proof*⟩

lemma *lim-Ima′′m*: $(\lambda n.\ \$(I\hat{}\{m\}a'')\hat{}\{m\}\text{-}n) \longrightarrow 1\ /\ (\$d\hat{}\{m\})\hat{}2$ **if** $m \neq 0$ $i > 0$ **for** *m*::*nat*
⟨*proof*⟩

lemma *perp-due-incr-calc*: $\$(I\hat{}\{m\}a'')\hat{}\{m\}\text{-}\infty = 1\ /\ (\$d\hat{}\{m\})\hat{}2$ **if** $m \neq 0$ $i > 0$ **for** *m*::*nat*
⟨*proof*⟩

end

end