```
THEORY Functions
  IMPORT THEORY PROJECTS
    /SimpleDEq THEORIES /SimpleDEq/Intervals.dtf|org.eventb.theory.core.deployedTheoryRoot#Intervals
  TYPE PARAMETERS E,F,G,H,I
  OPERATORS
    bind <expression> (fab: E ⇸ F,gac: E ⇸ G)
      direct definition
        (λ x · x ∈ dom(fab) ∩ dom(gac) | (fab(x) ↦ gac(x)))
    fproj1 <expression> (fa_bc: E ⇸ F×G)
      direct definition
        (λ x↦y · x ∈ F ∧ y ∈ G | x) ∘ fa_bc
    fproj2 <expression> (fa_bc: E ⇸ F×G)
      direct definition
        (λ x↦y · x ∈ F ∧ y ∈ G | y) ∘ fa_bc
    Rfplus <expression> ()
      direct definition
        (λ rf ↦ rg · rf ∈ (RReal → RReal) ∧ rg ∈ (RReal → RReal) |
          (λ x · x ∈ RReal | plus(rf(x) ↦ rg(x)))
        )
    Rftimes <expression> ()
      direct definition
        (λ rf ↦ rg · rf ∈ (RReal → RReal) ∧ rg ∈ (RReal → RReal) |
          (λ x · x ∈ RReal | times(rf(x) ↦ rg(x)))
        )
    Rfscal <expression> ()
      direct definition
        (λ l ↦ rf · l ∈ RReal ∧ rf ∈ (RReal → RReal) |
          (λ x · x ∈ RReal | times(l ↦ rf(x)))
        )
    Rfcste <expression> (l: RReal)
      direct definition
        (λ x · x ∈ RReal | l)
    partial1 <expression> (fab_c: E×F⇸G,y: F)
      direct definition
        (λ x · (x ↦ y) ∈ dom(fab_c) | fab_c(x↦y))
    partial2 <expression> (fab_c: E×F⇸G,x: E)
      direct definition
        (λ y · (x ↦ y) ∈ dom(fab_c) | fab_c(x↦y))
    partialComp <expression> (fabb: E×F⇸G,gab: E⇸F)
      direct definition
        (λ t · t ∈ dom(gab) ∧ (t ↦ gab(t)) ∈ dom(fabb) | fabb(t ↦ gab(t)))
    unpartialize1 <expression> (A: ℙ(E),fbc: F⇸G)
      direct definition
        (λ x↦y · x ∈ A ∧ y ∈ dom(fbc) | fbc(y))
    unpartialize2 <expression> (B: ℙ(F),fac: E ⇸ G)
      direct definition
        (λ x↦y · x ∈ dom(fac) ∧ y ∈ B | fac(x))
    increasing <predicate> (AR: ℙ(RReal),f: AR → RReal)
      direct definition
        ∀ x, y · x ∈ AR ∧ y ∈ AR ⇒ (
          (x ↦ y ∈ leq) ⇔ (f(x) ↦ f(y) ∈ leq)
        )
    strictlyIncreasing <predicate> (AR: ℙ(RReal),f: AR → RReal)
      direct definition
        ∀ x, y · x ∈ AR ∧ y ∈ AR ⇒ (
          (x ↦ y ∈ lt) ⇔ (f(x) ↦ f(y) ∈ lt)
        )
    decreasing <predicate> (AR: ℙ(RReal),f: AR → RReal)
      direct definition
        ∀ x, y · x ∈ AR ∧ y ∈ AR ⇒ (
          (x ↦ y ∈ leq) ⇔ (f(x) ↦ f(y) ∈ geq)
        )
    strictlyDecreasing <predicate> (AR: ℙ(RReal),f: AR → RReal)
      direct definition
        ∀ x, y · x ∈ AR ∧ y ∈ AR ⇒ (
          (x ↦ y ∈ lt) ⇔ (f(x) ↦ f(y) ∈ gt)
        )
    constant <predicate> (A: ℙ(E),fa: A → F)
      direct definition
        ∀ x · x ∈ A ⇒ (∀ y · y ∈ A ⇒ fa(x) = fa(y))
    positive <predicate> (A: ℙ(E),far: A → RReal)
      direct definition
        ∀ x · x ∈ A ⇒ Rzero ↦ far(x) ∈ leq
```

```
    strictlyPositive <predicate> (A: ℙ(E),far: A → RReal)
      direct definition
        ∀ x · x ∈ A ⇒ Rzero ↦ far(x) ∈ lt
    negative <predicate> (A: ℙ(E),far: A → RReal)
      direct definition
        ∀ x · x ∈ A ⇒ Rzero ↦ far(x) ∈ geq
    strictlyNegative <predicate> (A: ℙ(E),far: A → RReal)
      direct definition
        ∀ x · x ∈ A ⇒ Rzero ↦ far(x) ∈ gt
    application <expression> ()
      direct definition
        (λ f_ ↦ e_ · f_ ∈ E ⇸ F ∧ e_ ∈ dom(f_) | f_(e_))
    until <expression> (start: RReal,fre: RReal ⇸ E,t0: RReal,gre: RReal ⇸ E)
      well−definedness start ↦ t0 ∈ leq,Closed2Open(start,t0) ⊆ dom(fre),Closed2Infinity(t0) ⊆ dom(
          gre)
      direct definition
        (Closed2Open(start,t0) ◁ fre) ∪ (Closed2Infinity(t0) ◁ gre)
    untilF <expression> (start: RReal,fref: RReal×E ⇸ F,t0: RReal,gref: RReal×E ⇸ F)
      well−definedness start ↦ t0 ∈ leq,Closed2Open(start,t0)×∅ ⊆ dom(fref),Closed2Infinity(t0)×∅ ⊆
          dom(gref)
      direct definition
        ((Closed2Open(start,t0)×E) ◁ fref) ∪ ((Closed2Infinity(t0)×E) ◁ gref)
    fcste <expression> (A: ℙ(E),v: F)
      direct definition
        (λ x · x∈ A | v)
    boundedBy <predicate> (A: ℙ(E),far: E ⇸ RReal,fmin: RReal,fmax: RReal)
      well−definedness A ⊆ dom(far)
      direct definition
        ∀ x_ · x_ ∈ A ⇒ fmin ↦ far(x_) ∈ leq ∧ far(x_) ↦ fmax ∈ leq
  AXIOMATIC DEFINITIONS continuity:
    OPERATORS
      C0 <expression> (A: ℙ(E),B: ℙ(F)) : ℙ(ℙ(E×F))
        well−definedness A ≠ ∅,B ≠ ∅
      D1 <expression> (A2: ℙ(RReal),B: ℙ(F)) : ℙ(ℙ(RReal×F))
      Cn <expression> (n: ℕ,A: ℙ(E),B: ℙ(F)) : ℙ(ℙ(E×F))
      Dn <expression> (n: ℕ,A2: ℙ(RReal),B: ℙ(F)) : ℙ(ℙ(RReal×F))
        well−definedness n > 0
    AXIOMS
      cid:
        ∀ A,B · A ⊆ E ∧ B ⊆ F ⇒ (C0(A,B) = Cn(0,A,B))
      did:
        ∀ A,B · A ⊆ RReal ∧ B ⊆ F ⇒ (D1(A,B) = Dn(1,A,B))
      c_in_c:
        ∀ A,B,k · A ⊆ E ∧ B ⊆ F ∧ k ∈ ℕ ⇒ (Cn(k+1,A,B) ⊂ Cn(k,A,B))
      d_in_d:
        ∀ A,B,k · A ⊆ RReal ∧ B ⊆ F ∧ k ∈ ℕ ∧ k > 0 ⇒ (Dn(k+1,A,B) ⊂ Dn(k,A,B))
      c_in_d:
        ∀ A,B,k · A ⊆ RReal ∧ B ⊆ F ∧ k ∈ ℕ ∧ k > 0 ⇒ (Cn(k,A,B) ⊂ Dn(k,A,B))
      d_in_c:
        ∀ A,B,k · A ⊆ RReal ∧ B ⊆ F ∧ k ∈ ℕ ∧ k > 0 ⇒ (Dn(k,A,B) ⊂ Cn(k−1,A,B))
      c_inclusion_stable:
        ∀ A,C,B,k ·
          A ⊆ E ∧ C ⊆ A ∧ B ⊆ F ∧ k ∈ ℕ
            ⇒
              (Cn(k,A,B) ⊆ Cn(k,C,B))
      d_inclusion_stable:
        ∀ A,C,B,k ·
          A ⊆ RReal ∧ C ⊆ A ∧ B ⊆ F ∧ k ∈ ℕ ∧ k > 0
            ⇒
              (Dn(k,A,B) ⊆ Dn(k,C,B))
      d_restriction:
        ∀ A,C,B,k,f ·
          A ⊆ RReal ∧ C ⊆ A ∧ B ⊆ F ∧ k ∈ ℕ ∧ k > 0 ∧
          f ∈ RReal ⇸ B ∧ A ⊆ dom(f) ∧ f ∈ Dn(k,A,B) ⇒
            (C ◁ f) ∈ Dn(k,C,B) derivative:
    OPERATORS
      Der <expression> (A2: ℙ(RReal),B: ℙ(F),fa2b: RReal ⇸ B) : ℙ(RReal×F)
        well−definedness fa2b ∈ D1(A2,B),A2 ⊆ dom(fa2b)
      Dern <expression> (n: ℕ,A2: ℙ(RReal),B: ℙ(F),fa2b: RReal ⇸ B) : ℙ(RReal×F)
        well−definedness n > 0,A2 ⊆ dom(fa2b)
    AXIOMS
      derType:
        ∀ A,B,f · A ⊆ RReal ∧ B ⊆ F ∧ f ∈ (RReal ⇸ B) ∧ A ⊆ dom(f) ∧ f ∈ D1(A,B) ⇒ (Der(A,B,f) ∈ (A
            → F))
```

```
    derDef_1:
      ∀ A,B,f · A ⊆ RReal ∧ B ⊆ F ∧ f ∈ (RReal ⇸ B) ∧ A ⊆ dom(f) ∧ f ∈ Dn(1,A,B) ⇒ (
        Dern(1,A,B,f) = Der(A,B,f)
      )
    dernDef_n:
      ∀ A,B,f,k · A ⊆ RReal ∧ B ⊆ F ∧ f ∈ (RReal ⇸ B) ∧ A ⊆ dom(f) ∧ k ∈ ℕ ∧ k > 1 ∧ f ∈ Dn(k,A,B)
          ⇒ (
        Dern(k,A,B,f) = Der(A,B,Dern(k−1,A,B,f))
      )
    c_def:
      ∀ A,B,k,f · A ⊆ RReal ∧ B ⊆ F ∧ k ∈ ℕ ∧ k > 1 ∧ f ∈ (RReal ⇸ B) ∧ dom(f) ⊆ A ⇒ (
        (f ∈ Cn(k,A,B) ⇔ (f ∈ Dn(k,A,B) ∧ Dern(k,A,B,f) ∈ C0(A,B)))
      )  cartesian:
  AXIOMS
    cartesian_continuity:
      ∀ A,B,C,f,g,k ·
        A ⊆ E ∧ B ⊆ F ∧ C ⊆ G ∧ f ∈ (A → B) ∧ g ∈ (A → C) ∧ k ∈ ℕ ⇒
          (f ∈ Cn(k,A,B) ∧ g ∈ Cn(k,A,C) ⇔ (bind(f,g) ∈ Cn(k,A,B×C)))
    cartesian_derivability:
      ∀ A,B,C,f,g,k ·
        A ⊆ RReal ∧ B ⊆ F ∧ C ⊆ G ∧ f ∈ (A → B) ∧ g ∈ (A → C) ∧ k ∈ ℕ ∧ k > 0 ⇒
          (f ∈ Dn(k,A,B) ∧ g ∈ Dn(k,A,C) ⇔ (bind(f,g) ∈ Dn(k,A,B×C)))
    cartesian_der:
      ∀ A,B,C,f,g ·
        A ⊆ RReal ∧ B ⊆ F ∧ C ⊆ G ∧ f ∈ (A → B) ∧ g ∈ (A → C) ∧
        f ∈ D1(A,B) ∧ g ∈ D1(A,C) ⇒
          (Der(A,B×C,bind(f,g)) = bind(Der(A,B,f),Der(A,C,g)))
    cartesian_dern:
      ∀ A,B,C,f,g,k ·
        A ⊆ RReal ∧ B ⊆ F ∧ C ⊆ G ∧ f ∈ (A → B) ∧ g ∈ (A → C) ∧ k ∈ ℕ ∧ k > 0 ∧
        f ∈ Dn(k,A,B) ∧ g ∈ Dn(k,A,C) ⇒
          (Dern(k,A,B×C,bind(f,g)) = bind(Dern(k,A,B,f),Dern(k,A,C,g)))  partials:
  AXIOMS
    partial_cue:
      ∀ A,B,C,f,k ·
        A ⊆ E ∧ B ⊆ F ∧ C ⊆ G ∧ f ∈ (A×B→C) ∧ k ∈ ℕ ⇒ (
          f ∈ Cn(k,A×B,C) ⇔ ((∀ y0 · y0 ∈ B ⇒ partial1(f,y0) ∈ Cn(k,A,C)) ∧ (∀ x0 · x0 ∈ A ⇒
              partial2(f,x0) ∈ Cn(k,B,C)))
      )  real_functions:
  AXIOMS
    cste_cue:
      ∀ l · l ∈ RReal ⇒ (∀ k · k ∈ ℕ ⇒ (Rfcste(l) ∈ Cn(k,RReal,RReal)))
    cste_der:
      ∀ l · l ∈ RReal ⇒ (∀ k · k ∈ ℕ ∧ k > 0 ⇒ (Rfcste(l) ∈ Dn(k,RReal,RReal)))
    cste_der_def:
      ∀ l · l ∈ RReal ⇒ (Der(RReal,RReal,Rfcste(l)) = Rfcste(Rzero))
    plus_cue:
      ∀ A,f,g,k ·
        A ⊆ RReal ∧
        f ∈ (A → RReal) ∧ g ∈ (A → RReal) ∧
        k ∈ ℕ ∧
        f ∈ Cn(k,A,RReal) ∧ g ∈ Cn(k,A,RReal) ⇒
          (Rfplus(f ↦ g) ∈ Cn(k,A,RReal))
    plus_der:
      ∀ A,f,g,k ·
        A ⊆ RReal ∧
        f ∈ (A → RReal) ∧ g ∈ (A → RReal) ∧
        k ∈ ℕ ∧ k > 0 ∧
        f ∈ Dn(k,A,RReal) ∧ g ∈ Dn(k,A,RReal) ⇒
          (Rfplus(f ↦ g) ∈ Dn(k,A,RReal))
    plus_der_def:
      ∀ A,f,g,k ·
        A ⊆ RReal ∧
        f ∈ (A → RReal) ∧ g ∈ (A → RReal) ∧
        k ∈ ℕ ∧ k > 0 ∧
        f ∈ Dn(k,A,RReal) ∧ g ∈ Dn(k,A,RReal) ⇒
          (Dern(k,A,RReal,Rfplus(f ↦ g)) = Rfplus(Dern(k,A,RReal,f) ↦ Dern(k,A,RReal,g)))
    times_cue:
      ∀ A,f,g,k ·
        A ⊆ RReal ∧
        f ∈ (A → RReal) ∧ g ∈ (A → RReal) ∧
        k ∈ ℕ ∧
        f ∈ Cn(k,A,RReal) ∧ g ∈ Cn(k,A,RReal) ⇒
          (Rftimes(f ↦ g) ∈ Cn(k,A,RReal))
```

```
      times_der :
        ∀ A, f , g , k ·
          A ⊆ RReal ∧
          f ∈ (A → RReal) ∧ g ∈ (A → RReal) ∧
          k ∈ ℕ ∧ k > 0 ∧
          f ∈ Dn(k ,A, RReal) ∧ g ∈ Dn(k ,A, RReal) ⇒
            (Rftimes(f ↦ g) ∈ Dn(k ,A, RReal))
      times_der_def :
        ∀ A, f , g ·
          A ⊆ RReal ∧
          f ∈ (A → RReal) ∧ g ∈ (A → RReal) ∧
          f ∈ D1(A, RReal) ∧ g ∈ D1(A, RReal) ⇒
            (Der(A, RReal , Rftimes(f ↦ g)) = Rfplus(Rftimes(Der(A, RReal , f) ↦ g) ↦ Rftimes(f ↦ Der(A,
                RReal , g))))
      scal_cue :
        ∀ A, l , f , k ·
          A ⊆ RReal ∧
          l ∈ RReal ∧ f ∈ (A → RReal) ∧
          k ∈ ℕ ∧
          f ∈ Cn(k ,A, RReal) ⇒
            (Rfscal(l ↦ f) ∈ Cn(k ,A, RReal))
      scal_der :
        ∀ A, l , f , k ·
          A ⊆ RReal ∧
          l ∈ RReal ∧ f ∈ (A → RReal) ∧
          k ∈ ℕ ∧ k > 0 ∧
          f ∈ Dn(k ,A, RReal) ⇒
            (Rfscal(l ↦ f) ∈ Dn(k ,A, RReal))
      scal_der_def :
        ∀ A, l , f , k ·
          A ⊆ RReal ∧
          l ∈ RReal ∧ f ∈ (A → RReal) ∧
          k ∈ ℕ ∧ k > 0 ∧
          f ∈ Dn(k ,A, RReal) ⇒
            (Dern(k ,A, RReal , Rfscal(l ↦ f)) = Rfscal(l ↦ Dern(k ,A, RReal , f)))
      comp_cue :
        ∀ A,B, f , g , k ·
          A ⊆ RReal ∧ B ⊆ RReal ∧
          f ∈ A → B ∧ g ∈ B → RReal ∧
          k ∈ ℕ ∧
          f ∈ Cn(k ,A,B) ∧ g ∈ Cn(k ,B, RReal) ⇒
            (g ∘ f ∈ Cn(k ,A, RReal))
      comp_der :
        ∀ A,B, f , g , k ·
          A ⊆ RReal ∧ B ⊆ RReal ∧
          f ∈ A → B ∧ g ∈ B → RReal ∧
          k ∈ ℕ ∧ k > 0 ∧
          f ∈ Dn(k ,A,B) ∧ g ∈ Dn(k ,B, RReal) ⇒
            (g ∘ f ∈ Dn(k ,A, RReal))
      comp_der_def :
        ∀ A,B, f , g ·
          A ⊆ RReal ∧ B ⊆ RReal ∧
          f ∈ A → B ∧ g ∈ B → RReal ∧
          f ∈ D1(A,B) ∧ g ∈ D1(B, RReal) ⇒
            (Der(A, RReal , g ∘ f) = Rftimes((Der(B, RReal , g) ∘ f) ↦ Der(A,B, f))) lipschitz :
    OPERATORS
      lipschitzContinuous <predicate> (A: ℙ(E) ,B: ℙ(F) , fab : A → B) :
      kLipschitzContinuous <predicate> (A: ℙ(E) ,B: ℙ(F) , fab : A → B, r : RReal) :
    AXIOMS
      klip_lip :
        ∀ A,B, f · A ⊆ E ∧ B ⊆ F ∧ f ∈ A → B ⇒ (
          lipschitzContinuous(A,B, f) ⇔ (∃ k · k ∈ RReal ⇒ kLipschitzContinuous(A,B, f , k))
        )
      lip_cue :
        ∀ A,B, f · A ⊆ E ∧ B ⊆ F ∧ f ∈ A → B ⇒ (
          lipschitzContinuous(A,B, f) ⇒ (f ∈ C0(A,B))
        )
      lip_inclusion_stable :
        ∀ A,B,C, f ·
          A ⊆ E ∧ B ⊆ F ∧ C ⊆ A ∧ f ∈ A → B ⇒ (
            lipschitzContinuous(A,B, f) ⇒ lipschitzContinuous(C,B, f)
          )
      klip_inclusion_stable :
        ∀ A,B,C, f , k ·
```

```
      A ⊆ E ∧ B ⊆ F ∧ C ⊆ A ∧ f ∈ A → B ∧ k ∈ RReal ⇒ (
        kLipschitzContinuous(A,B,f,k) ⇒ kLipschitzContinuous(C,B,f,k)
      ) trigo:
  OPERATORS
    pi <expression> () : RReal
    sin <expression> () : ℙ(RReal×RReal)
    cos <expression> () : ℙ(RReal×RReal)
  AXIOMS
    pi_pos:
      Rzero ↦ pi ∈ lt
    sin_dom:
      sin ∈ RReal → RReal
    cos_dom:
      cos ∈ RReal → RReal
    sin_bounded:
      boundedBy(RReal,sin,uminus(Rone),Rone)
    sin_ran:
      ran(sin) = Closed2Closed(uminus(Rone),Rone)
    cos_bounded:
      boundedBy(RReal,cos,uminus(Rone),Rone)
    cos_ran:
      ran(cos) = Closed2Closed(uminus(Rone),Rone)
    cos_cinf:
      ∀ n · n ∈ ℕ ⇒ cos ∈ Cn(n,RReal,RReal)
    sin_cinf:
      ∀ n · n ∈ ℕ ⇒ sin ∈ Cn(n,RReal,RReal)
    cos0:
      cos(Rzero) = Rone
    cospi:
      cos(pi) = uminus(Rone)
    cospi2:
      cos(divide(pi↦Rtwo)) = Rzero
    cosmpi2:
      cos(uminus(divide(pi↦Rtwo))) = Rzero
    sin0:
      cos(Rzero) = Rzero
    sinpi:
      cos(pi) = uminus(Rzero)
    sinpi2:
      cos(divide(pi↦Rtwo)) = Rone
    sinmpi2:
      cos(uminus(divide(pi↦Rtwo))) = uminus(Rone)
    sin_odd:
      ∀ x · x ∈ RReal ⇒ sin(uminus(x)) = uminus(sin(x))
    cos_even:
      ∀ x · x ∈ RReal ⇒ cos(uminus(x)) = cos(x)
    sin_refl_pi4:
      ∀ x · x ∈ RReal ⇒ sin(minus(divide(pi↦Rtwo) ↦ x)) = cos(x)
    cos_refl_pi4:
      ∀ x · x ∈ RReal ⇒ cos(minus(divide(pi↦Rtwo) ↦ x)) = sin(x)
    sin_refl_pi2:
      ∀ x · x ∈ RReal ⇒ sin(minus(pi ↦ x)) = sin(x)
    cos_refl_pi2:
      ∀ x · x ∈ RReal ⇒ cos(minus(pi ↦ x)) = uminus(cos(x))
    sin_2pi:
      ∀ x · x ∈ RReal ⇒ sin(plus(x ↦ times(Rtwo ↦ pi))) = sin(x)
    cos_2pi:
      ∀ x · x ∈ RReal ⇒ cos(plus(x ↦ times(Rtwo ↦ pi))) = cos(x) basic_topology:
  OPERATORS
    IsOpen <predicate> (A: ℙ(E)) :
  AXIOMS
    open_continuity:
      ∀ x,f,A,B ·
        x ∈ A ∧ A ⊆ E ∧ B ⊆ F ∧
        f ∈ E ⇸ F ∧ A ⊆ dom(f) ∧ f ∈ C0(A,F) ∧
        IsOpen(B) ∧ f(x) ∈ B ⇒
          (∃ y · y ∈ A ∧ f(y) ∈ B)
    open_continuity_R:
      ∀ x,f,A,B ·
        A ⊆ RReal ∧ B ⊆ F ∧
        x ∈ RReal ∧
        f ∈ RReal ⇸ F ∧ f ∈ C0(A,F) ∧
        IsOpen(B) ∧ f(x) ∈ B ⇒
          (∃ y · y ∈ A ∧ x ↦ y ∈ lt ∧ f(y) ∈ B)
```

```
THEOREMS
  functionEquality:
    ∀ f, g · f ∈ E ⇸ F ∧ g ∈ E ⇸ F ⇒ (
      (f = g) ⇔ (dom(f) = dom(g) ∧ (∀ x · x ∈ dom(f) ⇒ (f(x) = g(x))))
    )
  bind_type:
    ∀ A,B,C,f,g · A ⊆ E ∧ B ⊆ F ∧ C ⊆ G ∧ f ∈ A → B ∧ g ∈ A → C ⇒
      bind(f,g) ∈ A → B×C
  proj1_type:
    ∀ A,B,C,f · A ⊆ E ∧ B ⊆ F ∧ C ⊆ G ∧ f ∈ A → B×C ⇒ (
      fproj1(f) ∈ A → B
    )
  proj2_type:
    ∀ A,B,C,f · A ⊆ E ∧ B ⊆ F ∧ C ⊆ G ∧ f ∈ A → B×C ⇒ (
      fproj2(f) ∈ A → C
    )
  partial1_type:
    ∀ A,B,C,f,y · A ⊆ E ∧ B ⊆ F ∧ C ⊆ G ∧ f ∈ A×B → C ∧ y ∈ B ⇒ (
      partial1(f,y) ∈ A → C
    )
  partial2_type:
    ∀ A,B,C,f,x · A ⊆ E ∧ B ⊆ F ∧ C ⊆ G ∧ f ∈ A×B → C ∧ x ∈ A ⇒ (
      partial2(f,x) ∈ B → C
    )
  partialComp_type:
    ∀ A,B,B2,f,g ·
      A ⊆ E ∧ B ⊆ F ∧ B2 ⊆ G ∧
      f ∈ A×B → B2 ∧ g ∈ A → B ⇒
        partialComp(f,g) ∈ A → B2
  proj1_bind:
    ∀ A,B,C,f,g · A ⊆ E ∧ B ⊆ F ∧ C ⊆ G ∧ f ∈ A → B ∧ g ∈ A → C ⇒ (
      fproj1(bind(f,g)) = f
    )
  proj2_bind:
    ∀ A,B,C,f,g · A ⊆ E ∧ B ⊆ F ∧ C ⊆ G ∧ f ∈ A → B ∧ g ∈ A → C ⇒ (
      fproj2(bind(f,g)) = g
    )
  bind_proj:
    ∀ A,B,C,f · A ⊆ E ∧ B ⊆ F ∧ C ⊆ G ∧ f ∈ A → B×C ⇒ (
      bind(fproj1(f),fproj2(f)) = f
    )
  proj_cue:
    ∀ A,B,C,f,k · A ⊆ E ∧ B ⊆ F ∧ C ⊆ G ∧ f ∈ A → B×C ∧ k ∈ ℕ ⇒ (
      f ∈ Cn(k,A,B×C) ⇔ (fproj1(f) ∈ Cn(k,A,B) ∧ fproj2(f) ∈ Cn(k,A,C))
    )
  unpartialized1_cue:
    ∀ A,B,C,f,k · A ⊆ E ∧ B ⊆ F ∧ C ⊆ G ∧ f ∈ B → C ∧ k ∈ ℕ ∧ f ∈ Cn(k,B,C) ⇒
      (unpartialize1(A,f) ∈ Cn(k,A×B,C))
  unpartialized2_cue:
    ∀ A,B,C,f,k · A ⊆ E ∧ B ⊆ F ∧ C ⊆ G ∧ f ∈ A → C ∧ k ∈ ℕ ∧ f ∈ Cn(k,A,C) ⇒
      (unpartialize2(B,f) ∈ Cn(k,A×B,C))
  unpartialize2_partial1:
    ∀ A,B,C,f · A ⊆ E ∧ B ⊆ F ∧ C ⊆ G ∧ f ∈ A → C
      ⇒
        (∀ y · y ∈ B ⇒ partial1(unpartialize2(B,f),y) = f)
  unpartialize1_partial2:
    ∀ A,B,C,f · A ⊆ E ∧ B ⊆ F ∧ C ⊆ G ∧ f ∈ B → C
      ⇒
        (∀ x · x ∈ A ⇒ partial2(unpartialize1(A,f),x) = f)
  MeanValue:
    ∀ a,b,f,M ·
      a ∈ RReal ∧ b ∈ RReal ∧ a ↦ b ∈ lt ∧
      f ∈ RReal ⇸ RReal ∧ Closed2Closed(a,b) ⊆ dom(f) ∧ f ∈ D1(Closed2Closed(a,b),RReal) ∧
      M ∈ RReal ∧ (∀ x · x ∈ Closed2Closed(a,b) ⇒ (abs(Der(Closed2Closed(a,b),RReal,f)(x)) ↦ M ∈
          leq)) ⇒
        (abs(divide(minus(f(b) ↦ f(a)) ↦ minus(b ↦ a))) ↦ M ∈ leq)
  functionConcat:
    ∀ A, B, f, g ·
      A ⊆ E ∧ B ⊆ E ∧ f ∈ A → F ∧ g ∈ B → F ∧
      (∀ x · x ∈ (A ∩ B) ⇒ f(x) = g(x)) ⇒
        (f ∪ g) ∈ (A ∪ B) → F
  functionConcatTyping:
    ∀ f, g ·
      f ∈ E ⇸ F ∧ g ∈ E ⇸ F ∧
```

```
      (∀ x · x ∈ (dom(f) ∩ dom(g)) ⇒ f(x) = g(x)) ⇒
        (f ∪ g) ∈ (dom(f) ∪ dom(g)) → F
  restrictionTyping:
    ∀ A,B,f · A ⊆ E ∧ B ⊆ A ∧ f ∈ A → F ⇒ ((B ◁ f) ∈ B → F)
  restrictionDomain:
    ∀ f,A · f ∈ E ⇸ F ∧ A ⊆ E ⇒ dom(A ◁ f) = A
  partialFunction:
    ∀ A,B,f1,f2 ·
      A ⊆ E ∧ B ⊆ E ∧ A ∩ B = ∅ ∧
      f1 ∈ A → F ∧ f2 ∈ B → F ⇒ (
        (A ◁ (f1 ∪ f2)) = f1 ∧
        (B ◁ (f1 ∪ f2)) = f2
      )
  strictlyPositiveWeakening:
    ∀ A,f ·
      A ⊆ RReal ∧ f ∈ RReal ⇸ RReal ∧ A ⊆ dom(f) ∧
      strictlyPositive(A,f) ⇒
        positive(A,f)
  strictlyNegativeWeakening:
    ∀ A,f ·
      A ⊆ RReal ∧ f ∈ RReal ⇸ RReal ∧ A ⊆ dom(f) ∧
      strictlyNegative(A,f) ⇒
        negative(A,f)
  positiveDer2Increasing:
    ∀ A,f ·
      A ⊆ RReal ∧
      f ∈ A → RReal ∧ f ∈ D1(A,RReal) ⇒ (
        positive(A,Der(A,RReal,f))
        ⇔
          increasing(A,f)
      )
  strictlyPositiveDer2StrictlyIncreasing:
    ∀ A,f ·
      A ⊆ RReal ∧
      f ∈ RReal ⇸ RReal ∧ A ⊆ dom(f) ∧ f ∈ D1(A,RReal) ⇒ (
        strictlyPositive(A,Der(A,RReal,f))
        ⇔
          strictlyIncreasing(A,f)
      )
  negativeDer2Decreasing:
    ∀ A,f ·
      A ⊆ RReal ∧
      f ∈ A → RReal ∧ f ∈ D1(A,RReal) ⇒ (
        negative(A,Der(A,RReal,f))
        ⇔
          decreasing(A,f)
      )
  strictlyNegativeDer2StrictlyDecreasing:
    ∀ A,f ·
      A ⊆ RReal ∧
      f ∈ RReal ⇸ RReal ∧ A ⊆ dom(f) ∧ f ∈ D1(A,RReal) ⇒ (
        strictlyNegative(A,Der(A,RReal,f))
        ⇔
          strictlyDecreasing(A,f)
      )
  zeroDer2Constant:
    ∀ A,f ·
      A ⊆ RReal ∧
      f ∈ RReal ⇸ RReal ∧ A ⊆ dom(f) ∧ f ∈ D1(A,RReal) ⇒ (
        (∀ x · x ∈ A ⇒ Der(A,RReal,f)(x) = Rzero)
        ⇔
          constant(A,f)
      )
  meanValue_geq:
    ∀ a,b,m,f ·
      a ∈ RReal ∧ b ∈ RReal ∧ a ↦ b ∈ lt ∧ m ∈ RReal ∧
      f ∈ RReal ⇸ RReal ∧ Closed2Closed(a,b) ⊆ dom(f) ∧ f ∈ D1(Closed2Closed(a,b),RReal) ∧
      f(a) ↦ m ∈ geq ∧ positive(Closed2Closed(a,b),Der(Closed2Closed(a,b),RReal,f)) ⇒
        (∀ t · t ∈ Closed2Closed(a,b) ⇒ f(t) ↦ m ∈ geq)
  meanValue_geq_strict:
    ∀ a,b,m,f ·
      a ∈ RReal ∧ b ∈ RReal ∧ a ↦ b ∈ lt ∧ m ∈ RReal ∧
      f ∈ RReal ⇸ RReal ∧ Closed2Closed(a,b) ⊆ dom(f) ∧ f ∈ D1(Closed2Closed(a,b),RReal) ∧
      f(a) ↦ m ∈ geq ∧ strictlyPositive(Closed2Closed(a,b),Der(Closed2Closed(a,b),RReal,f)) ⇒
```

```
      (∀ t · t ∈ Open2Closed(a,b) ⇒ f(t) ↦ m ∈ gt)
meanValue_leq:
  ∀ a,b,m,f ·
    a ∈ RReal ∧ b ∈ RReal ∧ a ↦ b ∈ lt ∧ m ∈ RReal ∧
    f ∈ RReal ⇸ RReal ∧ Closed2Closed(a,b) ⊆ dom(f) ∧ f ∈ D1(Closed2Closed(a,b),RReal) ∧
    f(a) ↦ m ∈ leq ∧ negative(Closed2Closed(a,b),Der(Closed2Closed(a,b),RReal,f)) ⇒
      (∀ t · t ∈ Closed2Closed(a,b) ⇒ f(t) ↦ m ∈ leq)
meanValue_leq_strict:
  ∀ a,b,m,f ·
    a ∈ RReal ∧ b ∈ RReal ∧ a ↦ b ∈ lt ∧ m ∈ RReal ∧
    f ∈ RReal ⇸ RReal ∧ Closed2Closed(a,b) ⊆ dom(f) ∧ f ∈ D1(Closed2Closed(a,b),RReal) ∧
    f(a) ↦ m ∈ leq ∧ strictlyNegative(Closed2Closed(a,b),Der(Closed2Closed(a,b),RReal,f)) ⇒
      (∀ t · t ∈ Open2Closed(a,b) ⇒ f(t) ↦ m ∈ lt)
meanValue_positivity:
  ∀ a,b,f ·
    a ∈ RReal ∧ b ∈ RReal ∧ a ↦ b ∈ lt ∧
    f ∈ RReal ⇸ RReal ∧ Closed2Closed(a,b) ⊆ dom(f) ∧ f ∈ D1(Closed2Closed(a,b),RReal) ∧
    f(a) ↦ Rzero ∈ geq ∧ positive(Closed2Closed(a,b),Der(Closed2Closed(a,b),RReal,f)) ⇒
      positive(Closed2Closed(a,b),f)
meanValue_negativity:
  ∀ a,b,f ·
    a ∈ RReal ∧ b ∈ RReal ∧ a ↦ b ∈ lt ∧
    f ∈ RReal ⇸ RReal ∧ Closed2Closed(a,b) ⊆ dom(f) ∧ f ∈ D1(Closed2Closed(a,b),RReal) ∧
    f(a) ↦ Rzero ∈ leq ∧ negative(Closed2Closed(a,b),Der(Closed2Closed(a,b),RReal,f)) ⇒
      negative(Closed2Closed(a,b),f)
bind_bind_partialComp:
  ∀ f1,f2,x1,x2,Af,Bf,Cf,Ax,Bx,Cx ·
    Af ⊆ E ∧ Bf ⊆ F ∧ Cf ⊆ G ∧
    Ax ⊆ Af ∧ Bx ⊆ Bf ∧ Cx ⊆ Cf ∧
    f1 ∈ Af×(Bf×Cf) → Bf ∧ f2 ∈ Af×(Bf×Cf) → Cf ∧
    x1 ∈ Ax → Bx ∧ x2 ∈ Ax → Cx ⇒
      partialComp(
        bind(f1,f2),
        bind(x1,x2)
      ) = bind(
        partialComp(f1,bind(x1,x2)),
        partialComp(f2,bind(x1,x2))
      )
until_type:
  ∀ s,t0,f,g ·
    s ∈ RReal ∧ t0 ∈ RReal ∧ s ↦ t0 ∈ leq ∧
    f ∈ RReal ⇸ E ∧ g ∈ RReal ⇸ E ∧
    Closed2Open(s,t0) ⊆ dom(f) ∧ Closed2Infinity(t0) ⊆ dom(g) ⇒
      until(s,f,t0,g) ∈ Closed2Infinity(s) → E
until_restrict1:
  ∀ s,t0,f,g ·
    s ∈ RReal ∧ t0 ∈ RReal ∧ s ↦ t0 ∈ leq ∧
    f ∈ RReal ⇸ E ∧ g ∈ RReal ⇸ E ∧
    Closed2Open(s,t0) ⊆ dom(f) ∧ Closed2Infinity(t0) ⊆ dom(g) ⇒
      (Closed2Open(s,t0) ◁ until(s,f,t0,g)) = f
until_restrict2:
  ∀ s,t0,f,g ·
    s ∈ RReal ∧ t0 ∈ RReal ∧ s ↦ t0 ∈ leq ∧
    f ∈ RReal ⇸ E ∧ g ∈ RReal ⇸ E ∧
    Closed2Open(s,t0) ⊆ dom(f) ∧ Closed2Infinity(t0) ⊆ dom(g) ⇒
      (Closed2Infinity(t0) ◁ until(s,f,t0,g)) = g
until_bind1:
  ∀ s,t0,f1,f2,g ·
    s ∈ RReal ∧ t0 ∈ RReal ∧ s ↦ t0 ∈ leq ∧
    f1 ∈ RReal ⇸ E ∧ f2 ∈ RReal ⇸ E ∧ g ∈ RReal ⇸ F ∧
    Closed2Open(s,t0) ⊆ dom(f1) ∧ Closed2Infinity(t0) ⊆ dom(f2) ⇒ (
      bind(until(s,f1,t0,f2),g) = until(s,bind(f1,g),t0,bind(f2,g))
    )
until_bind2:
  ∀ s,t0,f,g1,g2 ·
    s ∈ RReal ∧ t0 ∈ RReal ∧ s ↦ t0 ∈ leq ∧
    f ∈ RReal ⇸ E ∧ g1 ∈ RReal ⇸ F ∧ g2 ∈ RReal ⇸ F ∧
    Closed2Open(s,t0) ⊆ dom(g1) ∧ Closed2Infinity(t0) ⊆ dom(g2) ⇒ (
      bind(f,until(s,g1,t0,g2)) = until(s,bind(f,g1),t0,bind(f,g2))
    )
untilF_type:
  ∀ s,t0,f,g ·
    s ∈ RReal ∧ t0 ∈ RReal ∧ s ↦ t0 ∈ leq ∧
    f ∈ RReal×E ⇸ F ∧ g ∈ RReal×E ⇸ F ∧
```

```
      Closed2Open(s,t0)×∅ ⊆ dom(f) ∧ Closed2Infinity(t0)×∅ ⊆ dom(g) ⇒
        untilF(s,f,t0,g) ∈ RReal×E ⇸ F
  untilF_type_strong:
    ∀ s,t0,f,g,A ·
      A ⊆ E ∧
      s ∈ RReal ∧ t0 ∈ RReal ∧ s ↦ t0 ∈ leq ∧
      f ∈ RReal×A → F ∧ g ∈ RReal×A → F ⇒
        untilF(s,f,t0,g) ∈ Closed2Infinity(s)×A → F
  untilF_bind1:
    ∀ s,t0,f1,f2,g ·
      s ∈ RReal ∧ t0 ∈ RReal ∧ s ↦ t0 ∈ leq ∧
      f1 ∈ RReal×E ⇸ F ∧ f2 ∈ RReal×E ⇸ F ∧ g ∈ RReal×E ⇸ G ∧
      Closed2Open(s,t0)×∅ ⊆ dom(f1) ∧ Closed2Infinity(t0)×∅ ⊆ dom(f2) ⇒ (
        bind(untilF(s,f1,t0,f2),g) = untilF(s,bind(f1,g),t0,bind(f2,g))
      )
  untilF_bind2:
    ∀ s,t0,f,g1,g2 ·
      s ∈ RReal ∧ t0 ∈ RReal ∧ s ↦ t0 ∈ leq ∧
      f ∈ RReal×E ⇸ F ∧ g1 ∈ RReal×E ⇸ G ∧ g2 ∈ RReal×E ⇸ G ∧
      Closed2Open(s,t0)×∅ ⊆ dom(g1) ∧ Closed2Infinity(t0)×∅ ⊆ dom(g2) ⇒ (
        bind(f,untilF(s,g1,t0,g2)) = untilF(s,bind(f,g1),t0,bind(f,g2))
      )
  fcste_cue:
    ∀ A,l · A ⊆ E ∧ l ∈ F ⇒ (fcste(A,l) ∈ C0(A,F))
  fcste_type:
    ∀ A,l · A ⊆ E ∧ l ∈ F ⇒ fcste(A,l) ∈ A → F
PROOF RULES
  typing:
  Metavariables
    A: ℙ(E)
    B: ℙ(F)
    C: ℙ(G)
    fab: ℙ(E×F)
    gac: ℙ(E×G)
    fa_bc: ℙ(E×(F×G))
    fab_c: ℙ(E×F×G)
    x: E
    y: F
    A2: ℙ(E)
  Rewrite Rules
    domainRestrictionInvolutive: A ◁ (A ◁ fab)
      rhs1: ⊤ ⇒ A ◁ fab
    domainRestrictionComplete: A ◁ fab
      rhs1: dom(fab) = A ⇒ fab
    basicDomainRestrictionTyping: (A ◁ fab) ∈ E ⇸ F
      rhs1: ⊤ ⇒ fab ∈ E ⇸ F
    basicBindTyping: bind(fab,gac) ∈ E ⇸ F×G
      rhs1: ⊤ ⇒ fab ∈ E ⇸ F ∧ gac ∈ E ⇸ G
    bindDomain: dom(bind(fab,gac))
      rhs1: ⊤ ⇒ dom(fab) ∩ dom(gac)
    fcsteTyping: fcste(A,y) ∈ A → F
      rhs1: ⊤ ⇒ ⊤
    bindDomSubset: A ⊆ dom(bind(fab,gac))
      rhs1: ⊤ ⇒ A ⊆ dom(fab) ∧ A ⊆ dom(gac)
    fproj1Typing: fproj1(fa_bc) ∈ E ⇸ F
      rhs1: ⊤ ⇒ fa_bc ∈ E ⇸ F×G
    fproj2Typing: fproj2(fa_bc) ∈ E ⇸ G
      rhs1: ⊤ ⇒ fa_bc ∈ E ⇸ F×G
    fproj1Domain: dom(fproj1(fa_bc))
      rhs1: ⊤ ⇒ dom(fa_bc)
    fproj2Domain: dom(fproj2(fa_bc))
      rhs1: ⊤ ⇒ dom(fa_bc)
  Inference Rules
    bindType: fab ∈ A → B, gac ∈ A → C ⊢ bind(fab,gac) ∈ A→B×C
    proj1Type: fa_bc ∈ A → B×C ⊢ fproj1(fa_bc) ∈ A → B
    proj2Type: fa_bc ∈ A → B×C ⊢ fproj2(fa_bc) ∈ A → C
    partial1Type: fab_c ∈ A×B → C, y ∈ B ⊢ partial1(fab_c,y) ∈ A → C
    partial2Type: fab_c ∈ A×B → C, x ∈ A ⊢ partial2(fab_c,x) ∈ B → C
    domainRestrictionType: A2 ⊆ A, fab ∈ A → B ⊢ (A2 ◁ fab) ∈ A2 → B
  binding:
  Metavariables
    A: ℙ(E)
    B: ℙ(F)
    C: ℙ(G)
```

```
    AR: ℙ(RReal)
    fab: ℙ(E×F)
    gac: ℙ(E×G)
    fab2: ℙ(E×F)
    gac2: ℙ(E×G)
    fa_bc: ℙ(E×(F×G))
    farb: ℙ(RReal×F)
    garc: ℙ(RReal×G)
    far_bc: ℙ(RReal×(F×G))
    n: ℤ
    x: E
  Rewrite Rules
    bindProjRew: bind(fproj1(fa_bc),fproj2(fa_bc))
      rhs1: ⊤ ⇒ fa_bc
    bindDerRew: Der(AR,B×C,bind(farb,garc))
      rhs1: ⊤ ⇒ bind(Der(AR,B,farb),Der(AR,C,garc))
    bindEquality: bind(fab,gac)=bind(fab2,gac2)
      rhs1: ⊤ ⇒ fab = fab2 ∧ gac = gac2
    bindC0Rew: bind(fab,gac) ∈ C0(A,B×C)
      rhs1: ⊤ ⇒ fab ∈ C0(A,B) ∧ gac ∈ C0(A,C)
    bindD1Rew: bind(farb,garc) ∈ D1(AR,B×C)
      rhs1: ⊤ ⇒ farb ∈ D1(AR,B) ∧ garc ∈ D1(AR,C)
    bindCnRew: bind(farb,garc) ∈ Cn(n,AR,B×C)
      rhs1: n ≥ 0 ⇒ farb ∈ Cn(n,AR,B) ∧ garc ∈ Cn(n,AR,C)
    bindDnRew: bind(farb,garc) ∈ Dn(n,AR,B×C)
      rhs1: n > 0 ⇒ farb ∈ Dn(n,AR,B) ∧ garc ∈ Dn(n,AR,C)
    bindRestrict: A ◁ bind(fab,gac)
      rhs1: ⊤ ⇒ bind(A ◁ fab, A ◁ gac)
    bindEvaulate: bind(fab,gac)(x)
      rhs1: ⊤ ⇒ fab(x)↦gac(x)
  continuity:
  Metavariables
    A: ℙ(E)
    B: ℙ(F)
    AR: ℙ(RReal)
    n: ℤ
  Rewrite Rules
    C0_to_Cn0: C0(A,B)
      rhs1: ⊤ ⇒ Cn(0,A,B)
    D1_toDn1: D1(AR,B)
      rhs1: ⊤ ⇒ Dn(1,AR,B)
    Cn0_to_C0: Cn(0,A,B)
      rhs1: ⊤ ⇒ C0(A,B)
    Dn1_to_D1: Dn(1,AR,B)
      rhs1: ⊤ ⇒ D1(AR,B)
  misc:
  Metavariables
    f: ℙ(E×F)
    g: ℙ(E×F)
  Rewrite Rules
    fun_equality: f = g
      rhs1: ⊤ ⇒ dom(f) = dom(g) ∧ (∀x · x ∈ dom(f) ⇒ f(x) = g(x))
  direct_product:
  Metavariables
    fab: ℙ(E×F)
    gac: ℙ(E×G)
  Rewrite Rules
    dirprodType: fab ⊗ gac ∈ E ⇸ F×G
      rhs1: ⊤ ⇒ fab ∈ E ⇸ F ∧ gac ∈ E ⇸ G
    dirprodDomain: dom(fab ⊗ gac)
      rhs1: ⊤ ⇒ dom(fab) ∩ dom(gac)
END
```