計算可能解析学と Coq

Holger Thies (京都大学)

2023 年 10 月 31 日

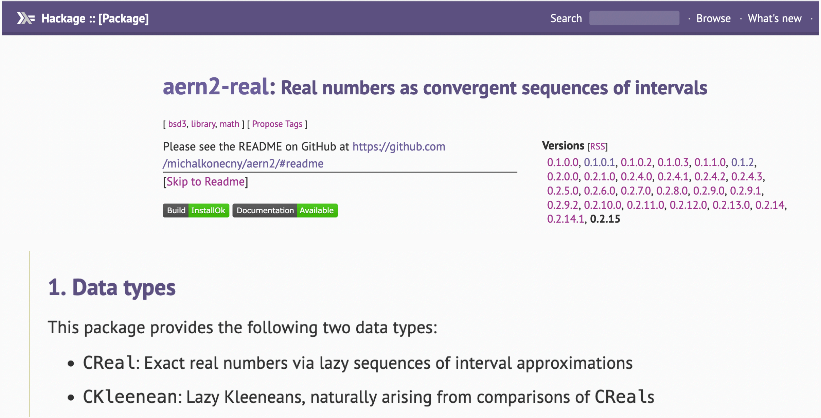
The 19th Theorem Proving and Provers meeting
2023 年 10 月 30 日-31 日
東京工業大学

This work is supported by KAKENHI grant numbers JP20K19744 and JP23H03346.

厳密実数計算と AERN

厳密実数計算（Exact Real Computation) とは誤差なしで実数の計算する方法である。

厳密実数計算のフレームワークはプログラミング言語に実数の型などを導入する。



Hackage :: [Package]   Search   · Browse · What's new ·

aern2-real: Real numbers as convergent sequences of intervals

[bsd3, library, math] [Propose Tags]

Please see the README on GitHub at https://github.com/michalkonecny/aern2/#readme

[Skip to Readme]

Build InstallOk   Documentation Available

Versions [RSS]
0.1.0.0, 0.1.0.1, 0.1.0.2, 0.1.0.3, 0.1.1.0, 0.1.2, 0.2.0.0, 0.2.1.0, 0.2.4.0, 0.2.4.1, 0.2.4.2, 0.2.4.3, 0.2.5.0, 0.2.6.0, 0.2.7.0, 0.2.8.0, 0.2.9.0, 0.2.9.1, 0.2.9.2, 0.2.10.0, 0.2.11.0, 0.2.12.0, 0.2.13.0, 0.2.14, 0.2.14.1, **0.2.15**

1. Data types

This package provides the following two data types:

- `CReal`: Exact real numbers via lazy sequences of interval approximations
- `CKleenean`: Lazy Kleeneans, naturally arising from comparisons of `CReal`s

AERN: 実数

実数の任意の正確な近似値の出力が可能

```
...> pi ? (bits 60)
[3.14159265358979323846264 21... ± 3.3087e-24 2^(-78)]
```

標準的な算術演算が利用可能

```
...> pi + pi*pi+2^(-3) ? (bits 60)
[13.13619705467915185729710 76... ± 5.0568e-23 2^(-74)]
```

AERN: 極限

早いコーシー列の極限が計算できる。

a に収束する早いコーシー列: $|\mathrm{a_n} - \mathrm{a}| \leq 2^{-\mathrm{n}}$

```
e_sum n = sum $ map (recip . fact) [0..n]
```

$\mathrm{e\_sum\ n} = \sum_{\mathrm{k}=0}^{\mathrm{n}} \frac{1}{\mathrm{k}!}$

e と *e_sum(n)* の差が $2^{-(\mathrm{n}-2)}$ より小さいため、
n -> e_sum(n+2) が e に収束する早いコーシー列である。

```
...> my_e = limit  \(n :: Integer) -> e_sum (n+2)
...> my_e ? (prec 1000)
[2.71828182845904523536028747... ±  ~0.0000 ~2^(-1217)]
```

AERN: 実数の比較

実数に対する比較演算子は、Kleenean（Lazy Boolean）のオブジェクトを返す。

```
...> pi > 0
{?(prec 36): CertainTrue}

...> pi == pi
{?(prec 36): TrueOrFalse}

...> pi == pi + 2^(-100)
{?(prec 36): TrueOrFalse}

...> (pi == pi + 2^(-100)) ? (prec 1000)
CertainFalse
```

- cAERN (Park, Konecný, T.) は公理的に厳密実数計算を定義する Coq ライブラリーである
- 一つの目標は AERN のユーザープログラムの抽出
- そのため公理的に AERN の型と演算子を導入する
  - K:true, false, $\perp$ が存在する Kleenean の型
  - R: 標準的な算術演算およびク Kleenean の比較演算子が存在する実数の型
  - 計算的に妥当な公理、例えば $\lim s : \forall(n, m).\ |s_n - s_m| < 2^{-n-m} \to \Sigma(r : \mathsf{R}).\ \forall(k : \mathsf{N}).\ |r - s_k| < 2^{-k}$

実数の公理

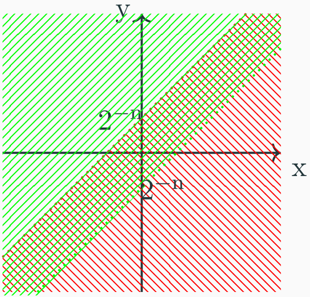
```
Class SemiDecOrderedField_Real (types : RealTypes) :=
  {
    mvmM :> MultivalueMonad_M types;
    real_0 : @ℝ types;
    real_1 : @ℝ types;
    real_plus : @ℝ types → @ℝ types → @ℝ types;
    real_mult : @ℝ types → @ℝ types → @ℝ types;
    real_opp : @ℝ types → @ℝ types;
    real_inv : ∀ {z}, z ≠ real_0 → @ℝ types;
    real_lt : @ℝ types → @ℝ types → ℙ;
    real_lt_semidec : ∀ x y : @ℝ types, semidec (real_lt x y);

    real_plus_comm : ∀ r1 r2 : @ℝ types, real_plus r1 r2 = real_plus r2 r1;
    real_plus_assoc : ∀ r1 r2 r3 : @ℝ types, real_plus (real_plus r1 r2) r3 = real_plus r1 (real_plus r2 r3);
    real_plus_inv : ∀ r : @ℝ types, real_plus r (real_opp r) = real_0;
    real_plus_unit : ∀ r : @ℝ types, real_plus real_0 r = r;
    real_mult_comm : ∀ r1 r2 : @ℝ types, real_mult r1 r2 = real_mult r2 r1;
    real_mult_assoc : ∀ r1 r2 r3 : @ℝ types, real_mult (real_mult r1 r2) r3 = real_mult r1 (real_mult r2 r3);
    real_mult_inv : ∀ (r : @ℝ types) (p : r ≠ real_0), real_mult (real_inv p) r = real_1;
    real_mult_unit : ∀ r : @ℝ types, real_mult real_1 r = r;
    real_mult_plus_distr: ∀ r1 r2 r3 : @ℝ types, real_mult r1 (real_plus r2 r3) = real_plus (real_mult r1 r2) (real_mult r1  r3);

    real_1_neq_0 : real_1 ≠ real_0;
    real_1_gt_0 : real_lt real_0 real_1;

    real_total_order : ∀ r1 r2 : @ℝ types, real_lt r1 r2 ∨ r1 = r2 ∨ real_lt r2 r1;
    real_lt_nlt : ∀ r1 r2 : @ℝ types, real_lt r1 r2 → ~ real_lt r2 r1;
    real_lt_lt_lt : ∀ r1 r2 r3 : @ℝ types, real_lt r1 r2 → real_lt r2 r3 → real_lt r1 r3;
    real_lt_plus_lt : ∀ r r1 r2 : @ℝ types, real_lt r1 r2 → real_lt (real_plus r r1) (real_plus r r2);
    real_lt_mult_pos_lt : ∀ r r1 r2 : @ℝ types, real_lt real_0 r → real_lt r1 r2 → real_lt (real_mult r r1) (real_mult r r2);
  }.
```

非決定性



Approximate splitting

softcomparison:$\forall(x, y : \mathrm{R}).\ \forall(n : \mathbb{N}).\ \mathsf{M}(x < y + 2^{-n} + y < x + 2^{-n})$

決定的な関数 $\mathrm{R} \to \mathrm{B}$ が存在しない。

非決定性

```
(* semideciability so that we can work on Prop directly, without mentioning K *)
Definition semidec := λ P : ℙ ⇒ {x : K | lazy_bool_up _ x ↔ P}.

Definition choose : ∀ p q, semidec p → semidec q → p ∨ q → ^M ({p}+{q}).
Proof.
  intros.
  destruct X.
  destruct X0.
  destruct i.
  destruct i0.
  apply (M_lift ({lazy_bool_up _ x} + {lazy_bool_up _ x0})).
  intro.
  destruct H4; auto.
  apply select.
  destruct H; auto.
Defined.
```

例：実数の最大値

$\mathrm{max(x, y)}$ を以下の（非決定的な）列の極限にする

- $\mathrm{If\ x < y + 2^{-n},\ then\ f(n) = y}$
- $\mathrm{If\ y < x + 2^{-n},\ then\ f(n) = x}$

このように定義されている全ての列が以下の形である。

$$\mathrm{x, x, y, y, x, x, x, y, x, y, y, y, y, y, y, y, y, y, y}$$

プログラム抽出

$$\forall(\mathrm{x}:\mathrm{R}).\ \mathsf{M}\Sigma(\mathrm{y}:\mathrm{R}).\ \mathrm{P}\ \mathrm{x}\ \mathrm{y}$$

の証明から入力 x に対して P を満たす実数 y を計算するプログラムが抽出可能である。

そのため Coq で定義された型 R は AERN の *CReal*、実数の演算子は AERN の演算子に抽出する。

抽出の例

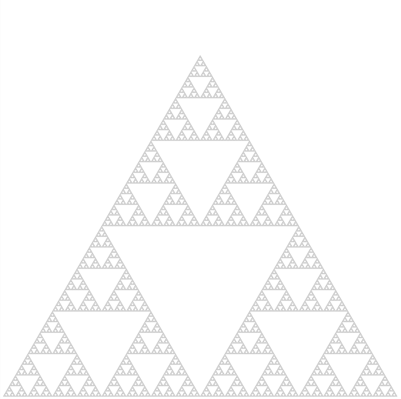
```
Lemma real_max_prop :
  forall x y, {z | (x >= y → z = x)
                   ∧ (x < y → z = y)}.
  Proof.
    intros.
    apply real_mslimit_P_lt.
    + (* max is single-valued *)
      ...
    + (* construct limit *)
      intros.
      apply (mjoin (x>y − prec n)
                   (y>x − prec n)).
      ++ intros [c1|c2].
         +++ (* when x > y − 2^-n *)
         exists x.
         ...
         +++ (* when x < y − 2^-n *)
         exists y.
         ...
      ++ apply M_split.
         apply prec_pos.
  Defined.
```

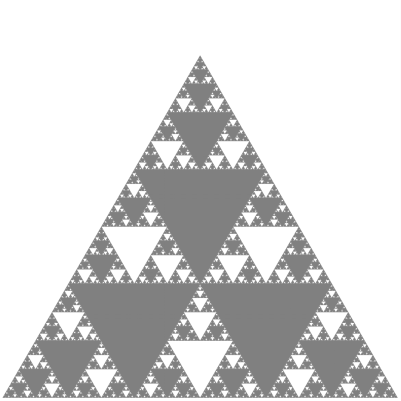
```
real_max_prop ::
    AERN2.CReal ->
    AERN2.CReal ->
    AERN2.CReal
real_max_prop x y =
  AERN2.limit (\n ->
    Prelude.id (\h -> case h of {
                        P.True -> x;
                        P.False -> y})
   (m_split x y (prec n)))
```


```
Lemma real_max_prop :
  forall x y, {z | (x >= y → z = x)
                   ∧ (x < y → z = y)}.
```

```
real_max_prop ::
    AERN2.CReal ->
    AERN2.CReal ->
```

Example: Fractal drawings





結論と今後の課題

- 実数、複素数上の証明から効率的な AERN プログラムを抽出できます。
- 部分集合や関数空間を扱う際、それらを効率的に表現することが重要です。
- これからは、zooming に基づくより便利な部分集合表現を実践する予定。
- また、関数空間につきまして、Taylor モデルに基づくより効率的な表現を実践する予定。

ご清聴ありがとうございました!

References

Github repository:
http://www.github.com/holgerthies/coq-aern/

Michal Konený, Sewon Park, and Holger Thies.
Certified Computation of Nondeterministic Limits.
NASA Formal Methods, 14th International Symposium, 2022.

Michal Konený, Sewon Park, and Holger Thies.
Extracting efficient exact real number computation from proofs in constructive type theory.
arxiv preprint, 2022.

Michal Konený, Sewon Park, and Holger Thies.
Formalizing Hyperspaces for Extracting Efficient Exact Real Computation.
MFCS 2023.