

(Haskellによる)

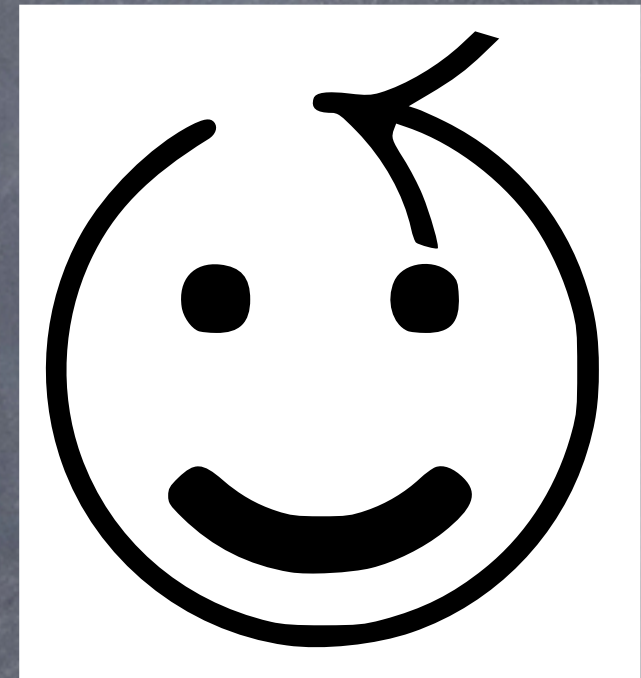
代数的実数とCADの実装紹介

2013-05-04 Haskellで計算機代数勉強会

酒井政裕

自己紹介

- 今回の勉強会の首謀者
- ブログ: ヒビルテ
- Twitter: @masahiro_sakai
- 定理証明や決定手続きに興味
 - github.com/msakai/toysolver
 - 今日はその中から、代数的実数の実装と、実閉体の理論の決定可能性について紹介



CM: Alloy本&TAPL好評発売中

抽象によるソフトウェア設計 Alloy ではじめる形式手法

Daniel Jackson 著

中島 豊 監訳

今井 健男・酒井 政裕・遠藤 佑介・片岡 欣夫 共訳

Software Abstractions
Logic, Language, and Analysis



型システム入門 プログラミング言語と型の理論

Benjamin C. Pierce 著

Types and Programming Languages

住井英二郎 監訳

遠藤佑介・酒井政裕・今井敬吾・黒木裕介・今井宣洋・才川隆文・今井健男 共訳



今日の話

- 話す内容

- 前半: 代数的実数の話

- 後半: 実閉体の理論の決定可能性

- 数学的に高度なことは特に使いません。

自分自身、代数に詳しいわけではないし (^_^;

前半：代数的実数編

目次

- ① 代数的実数とは
- ② 代数的実数の表現
- ③ 根の求め方
- ④ 代数的実数の演算
- ⑤ 課題と今後

代数的実数とは

- 有理数係数多項式の根になっている実数

- 例

- $\sqrt{2}$ と $-\sqrt{2}$ は $x^2 - 2$ の根 (i.e. $x^2 - 2 = 0$ の解)

- $\sqrt{2} + \sqrt{3} \doteq 3.1462643699419\dots$ は $x^4 - 10x^2 + 1$ の根

- (冪根 + 四則演算で書けるとは限らないので注意)

- 代数的実数でないもの

- 複素数: i は $x^2 + 1$ の根だけれど実数ではないのでNG

- 超越数: π とか e は有理数係数多項式の根では表現不可

なぜ代数的実数?

- 多項式制約の解を表現するのに使いたい
- 豊かな演算
 - 四則演算、多項式の根に関して閉じている
- 計算可能性
 - 演算や比較がすべて計算可能
 - 有理数と同じく数値誤差無しに計算可能
 - 数値計算なら誤差があっても良いけど、定理証明などでは誤差なしで計算したい

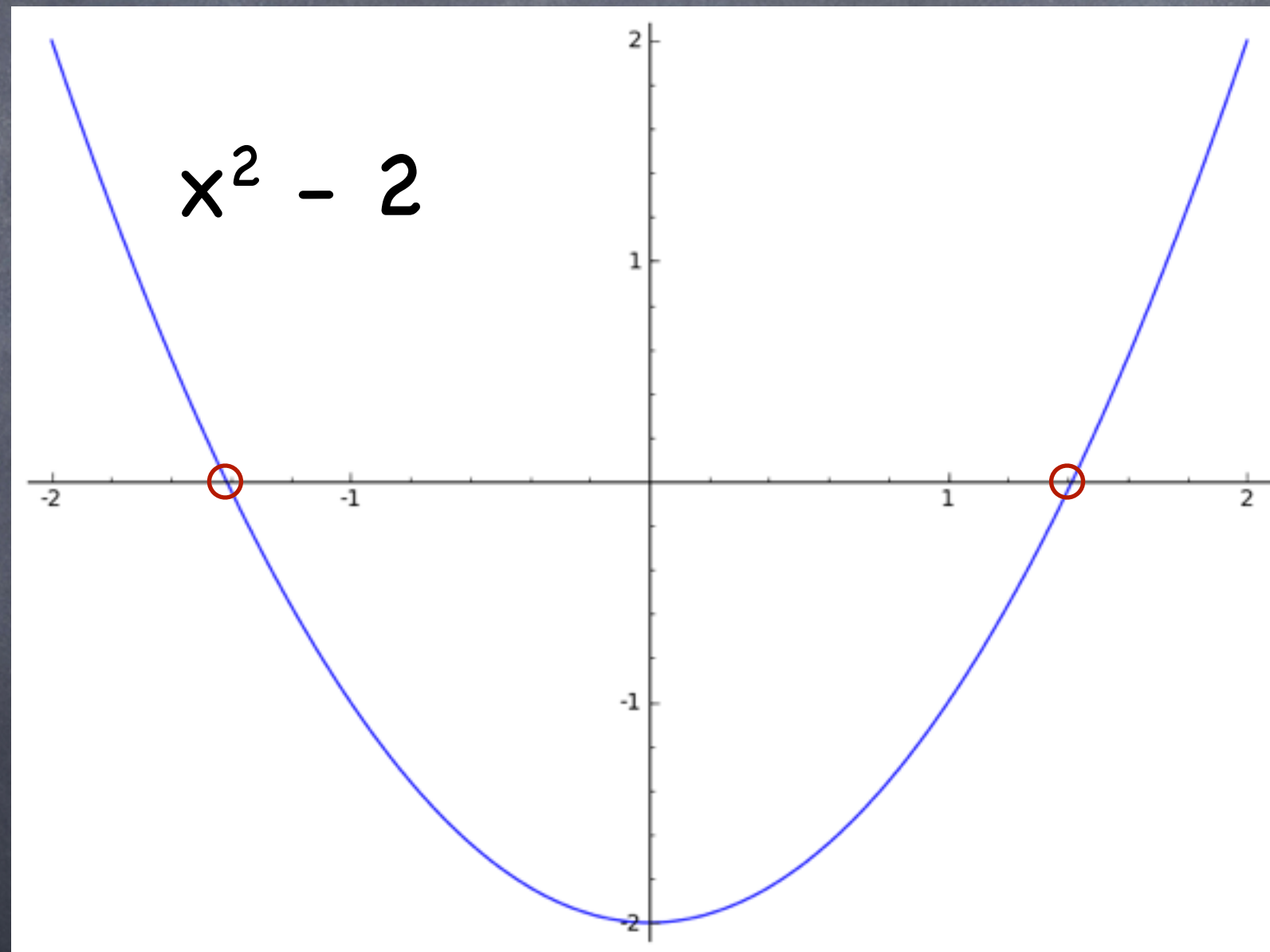
目次

- ◉ 代数的実数とは
- ◉ 代数的実数の表現
- ◉ 根の求め方
- ◉ 代数的実数の演算
- ◉ 課題と今後

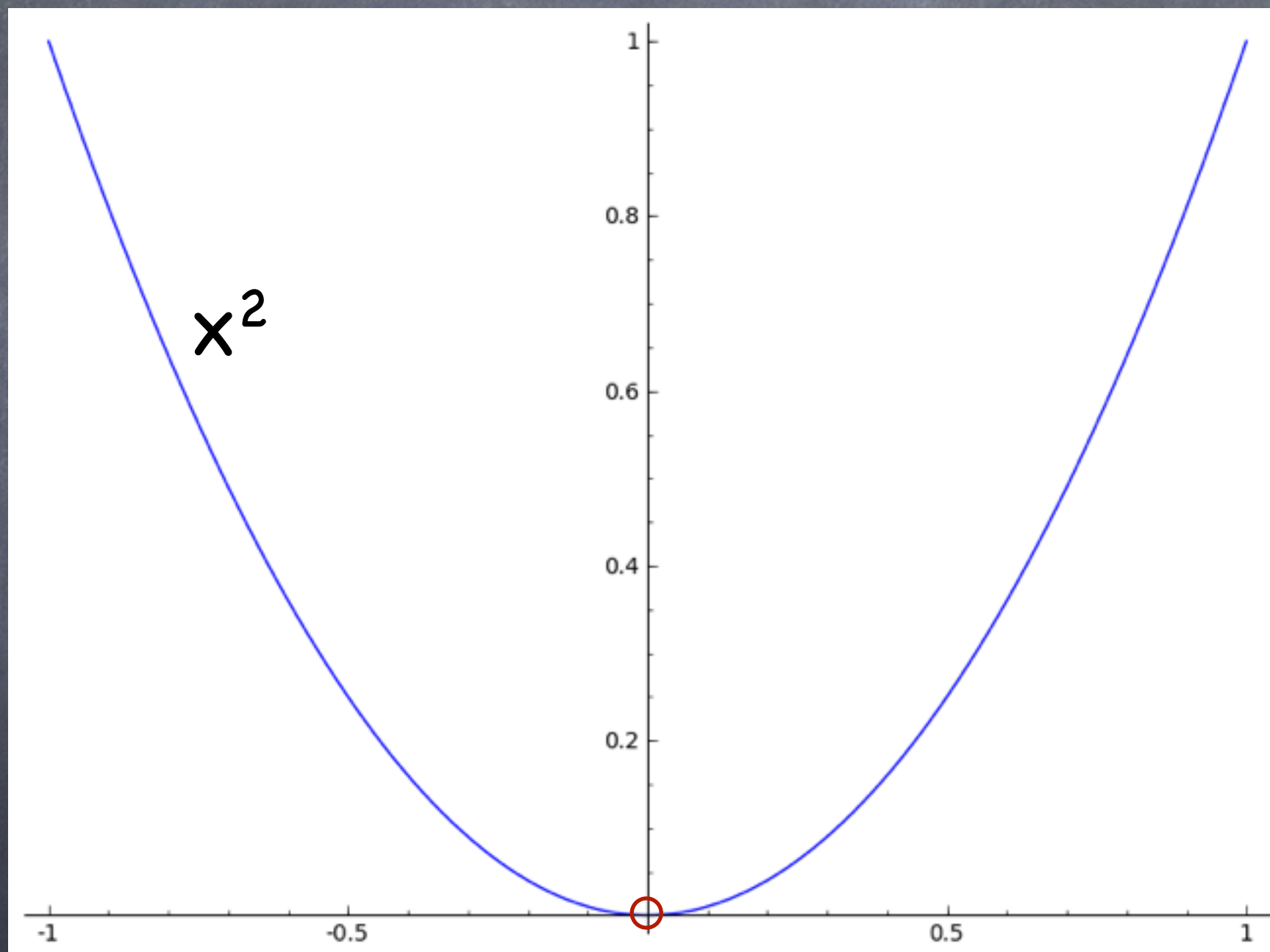
代数的実数の表現

- 多項式の根は一般に複数
 - n 次多項式は複素数の範囲で、重根を含めて n 個の根を持つ
 - 今回は実根に限定してるのと、重根を除くと、より少なくなるけれど
- 多項式だけでは根を特定できない！

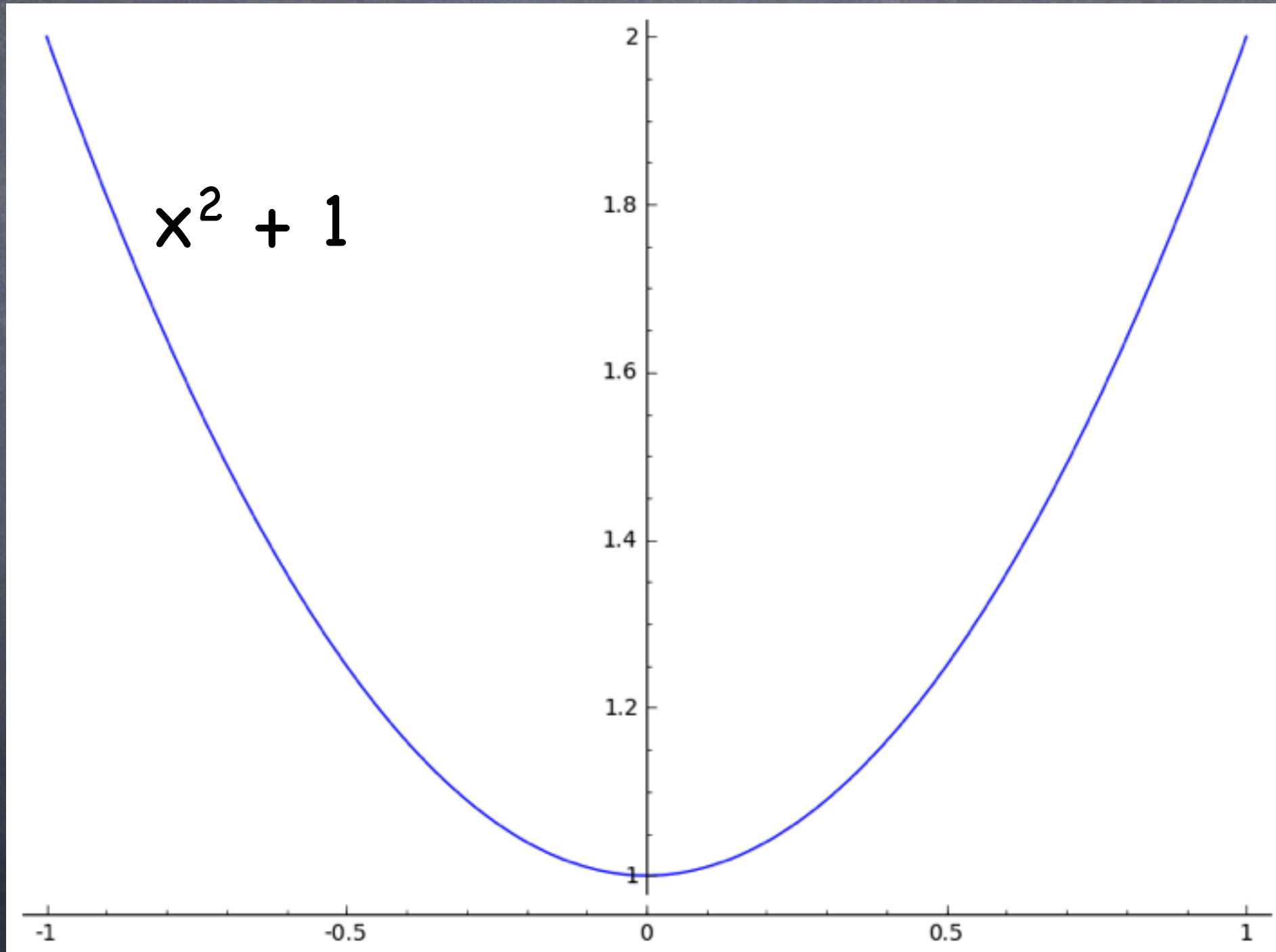
ちょうど n 個の根の場合



重根がある場合

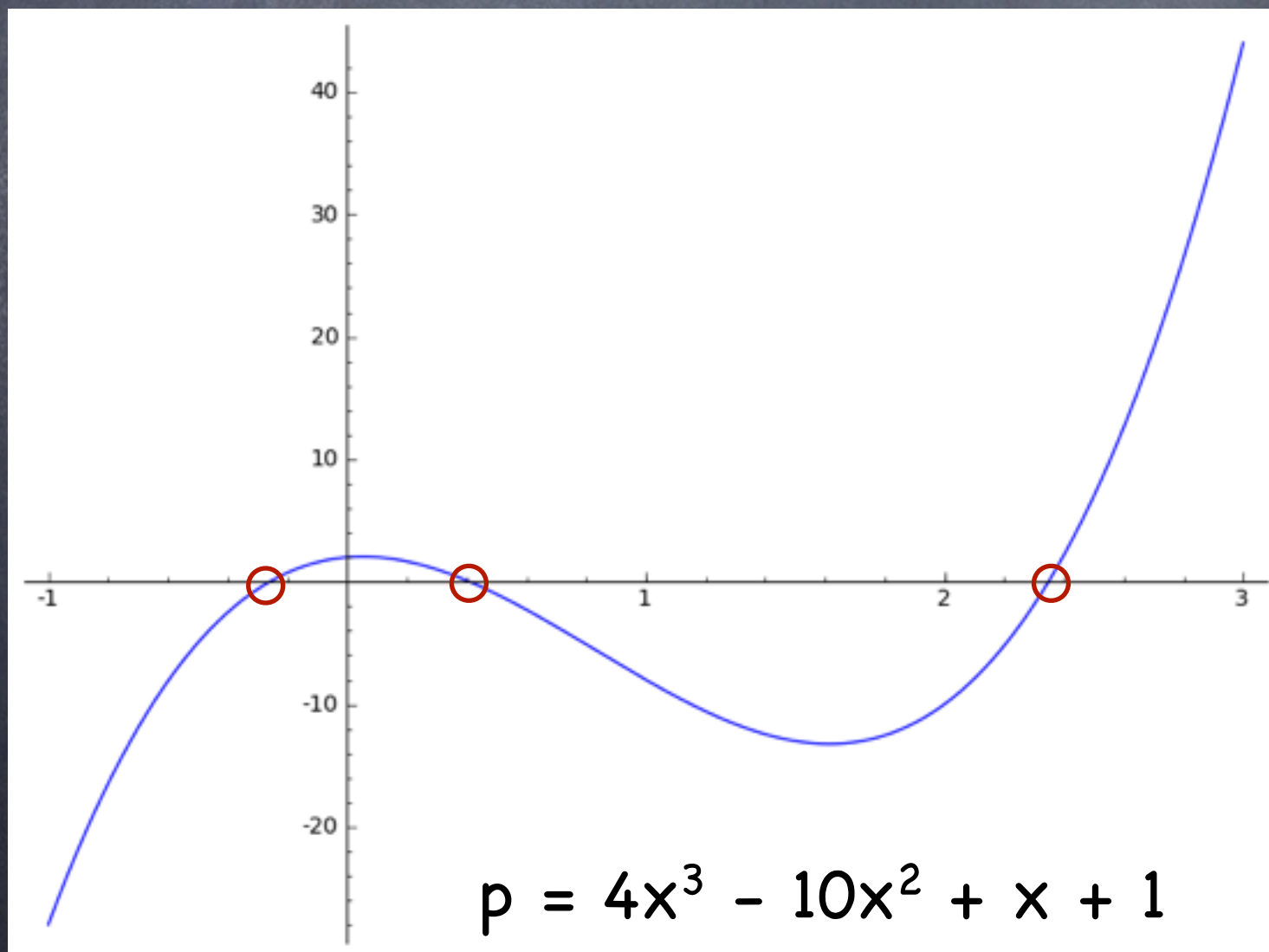


実根がない場合



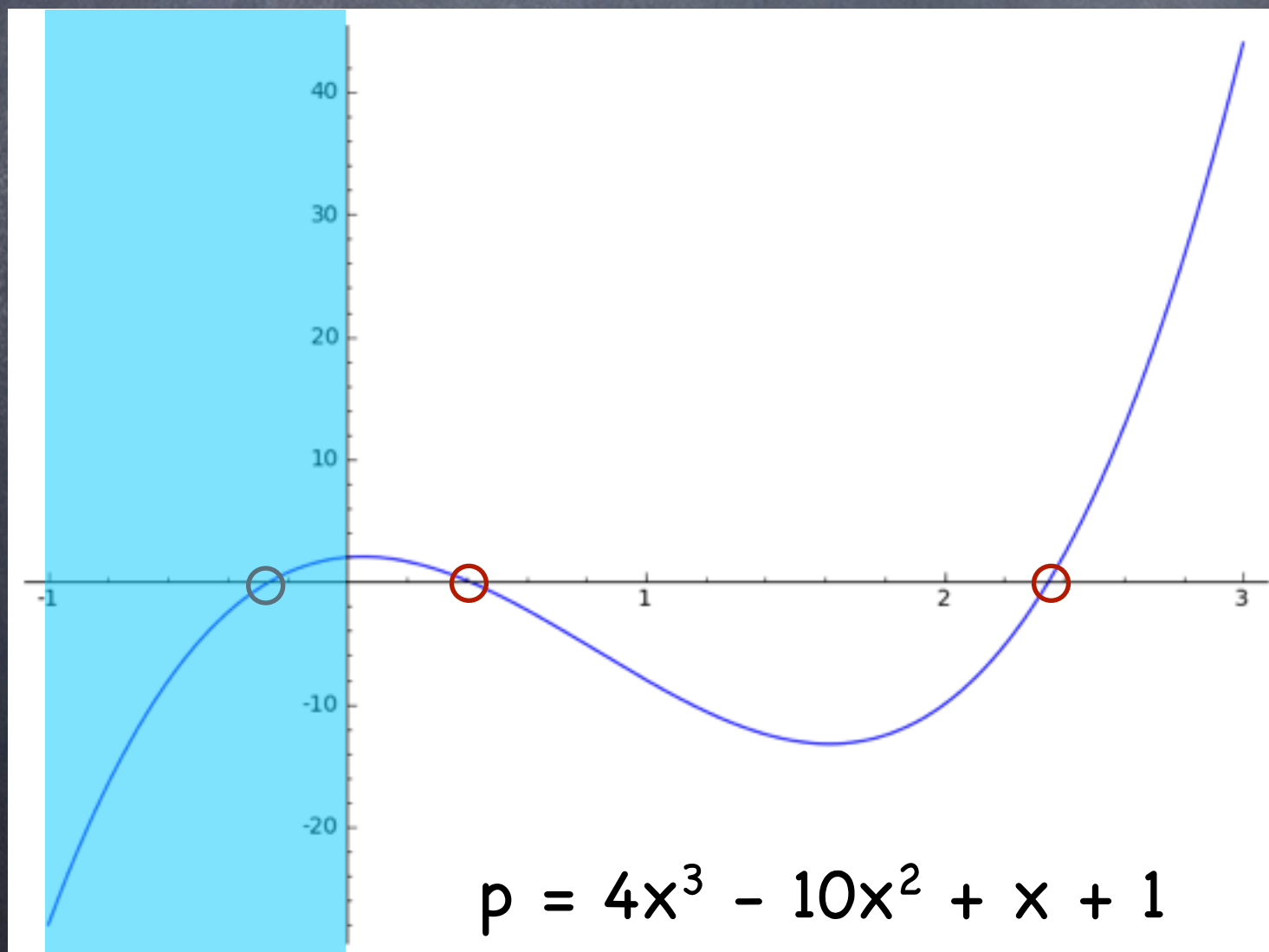
代数的実数の表現

- 多項式の根は複数なので、多項式だけでは根を特定できない！
- アイディア: 多項式 $\times$ 根をただ1つ含む区間



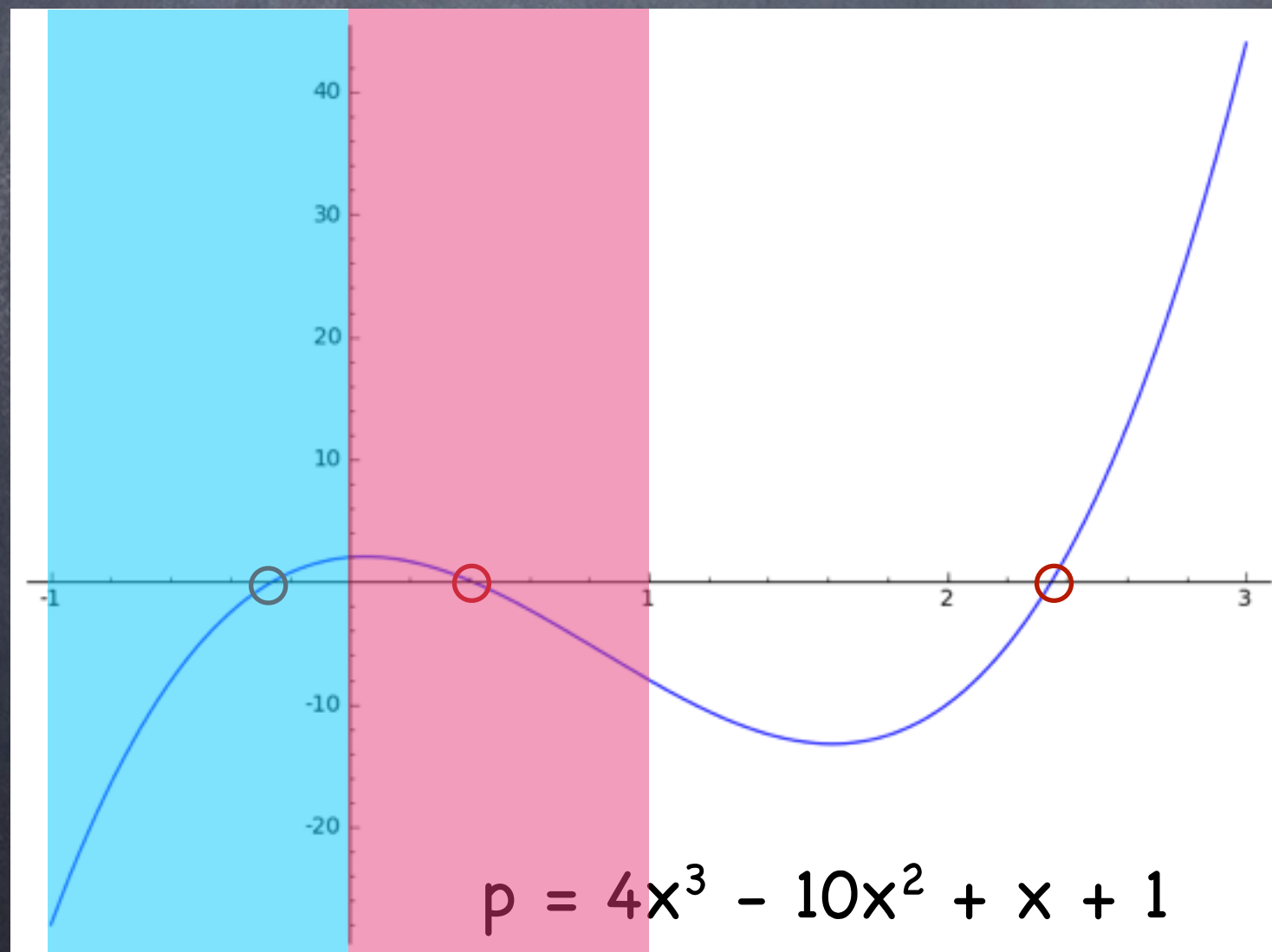
代数的実数の表現

- 多項式の根は複数なので、多項式だけでは根を特定できない！
- アイディア: 多項式 $\times$ 根をただ1つ含む区間



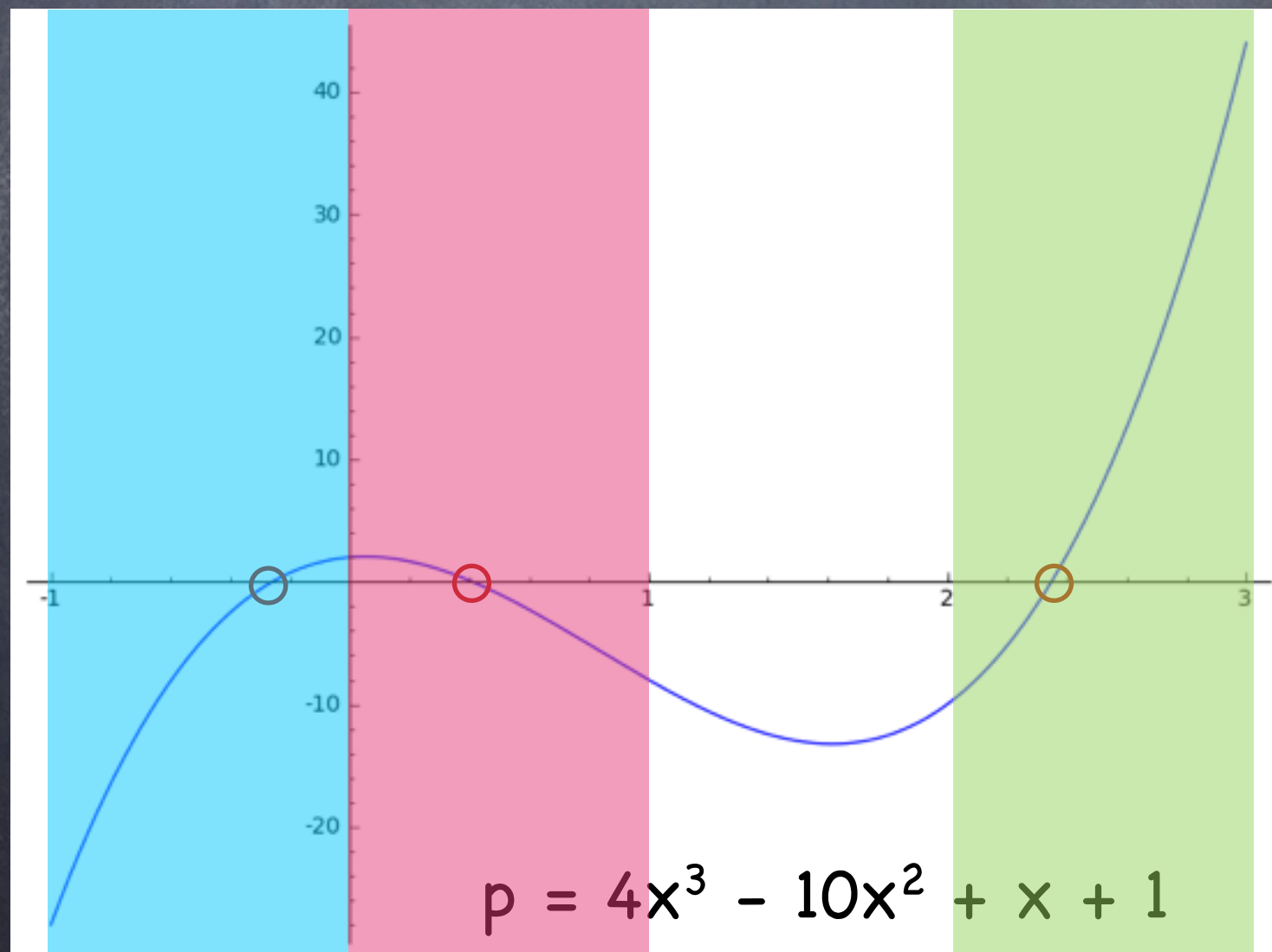
代数的実数の表現

- 多項式の根は複数なので、多項式だけでは根を特定できない！
- アイディア: 多項式 $\times$ 根をただ1つ含む区間



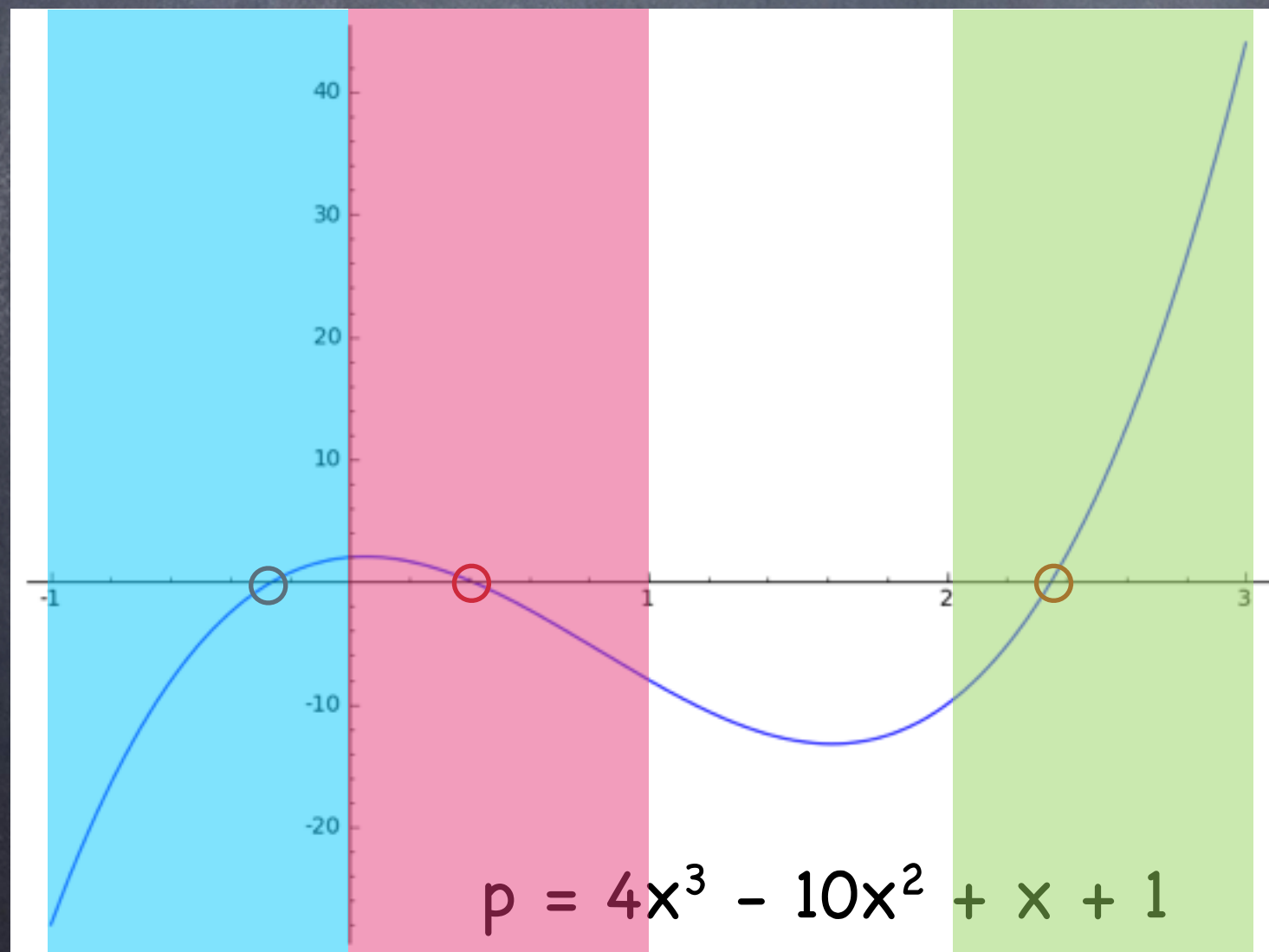
代数的実数の表現

- 多項式の根は複数なので、多項式だけでは根を特定できない！
- アイディア: 多項式 $\times$ 根をただ1つ含む区間



代数的実数の表現

- 多項式の根は複数なので、多項式だけでは根を特定できない！
- アイディア: 多項式 $\times$ 根をただ1つ含む区間



根の表現

1. $(p, [-1, 0))$
2. $(p, [0, 1])$
3. $(p, [2, 3])$

代数的実数の表現

- 自分の現在の実装での表現
 - `data AReal =`
 `AReal (UPolynomial Rational) (Interval Rational)`
- 有理数係数の最小多項式
 - 因数分解済み・最高次の係数を1に正規化
- その多項式の根をただ1つ含む有理数区間
 - (区間の実装は`data-interval`としてHackageに登録)
- 表現例
 - $-\sqrt{2} = \text{AReal } (x^2 - 2) (-2 \leq \cdot \leq 0)$
 - $\sqrt{2} = \text{AReal } (x^2 - 2) (0 \leq \cdot \leq 2)$

代数的実数の表現

- 別の表現方法

- 区間の代わりに根に番号をふって、番号との組で表現

- 代数的実数係数の多項式を利用

- 再帰的な表現

- 利点: 表現はコンパクトになる

- 欠点: 操作が面倒

目次

- ◉ 代数的実数とは
- ◉ 代数的実数の表現
- ◉ 根の求め方
- ◉ 代数的実数の演算
- ◉ 課題と今後

根を求める

- 根を求める関数
`realRoots :: UPolynomial Rational → [AResult]`
- お仕事
 - 多項式を因数分解
 - 因数分解して得られた各多項式について
 - すべての根を含む区間を計算 (根の限界)
 - 区間の分割を繰り返して、各区間が根をただ1つだけ含むようにする (根の分離)

根の限界

● 例)

$$4x^3 - 10x^2 + x + 1 = x^3 + (-5/2)x^2 + (1/4)x + (1/4)$$

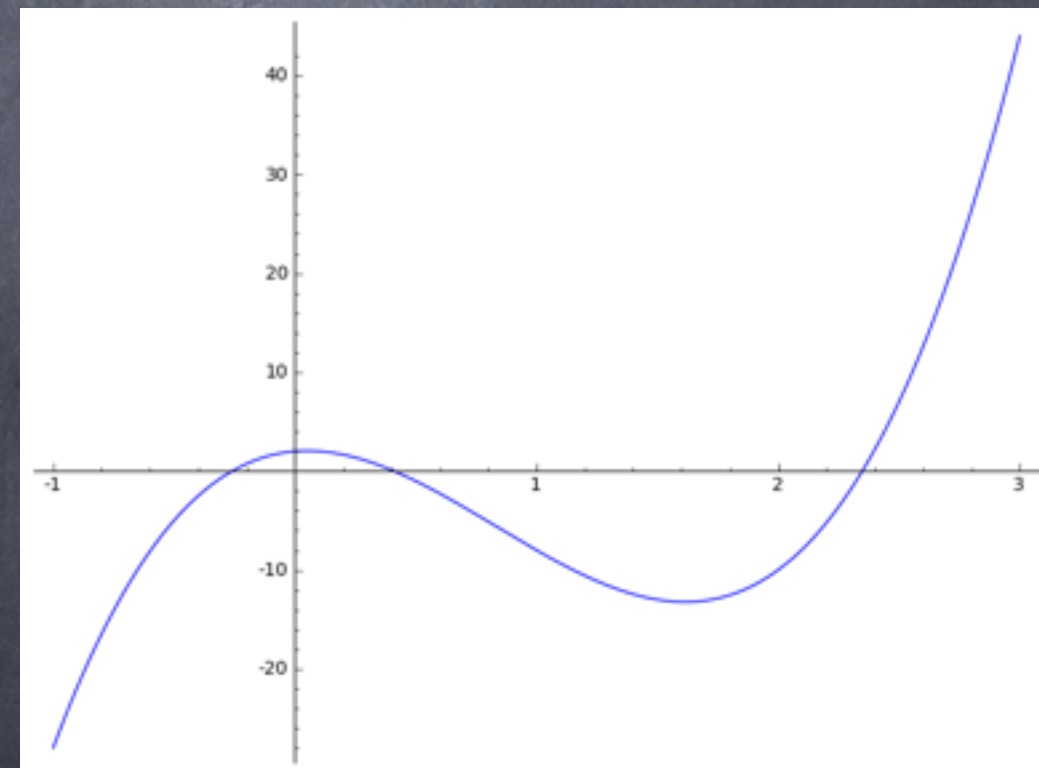
$$M = \max\{ 1, |-5/2| + |1/4| + |1/4| \} = 3$$

● まず全ての根を含む
有限の区間を計算

● [定理]

$x^n + a_{n-1}x^{n-1} + \dots + a_0$ の実根は、

$M = \max\{ 1, |a_{n-1}| + \dots + |a_0| \}$
とおくと $[-M, M]$ に含まれる



根の分離

- スツルム鎖

$p_0, p_1, p_2, \dots, p_m$

- $p_0 = p$

- $p_1 = p'$

- $p_2 = -(p_0 \bmod p_1)$

- ...

- $p_m = -(p_{m-2} \bmod p_{m-1})$

- $0 = -(p_{m-1} \bmod p_m)$

- (p, p') のGCD計算の変形)

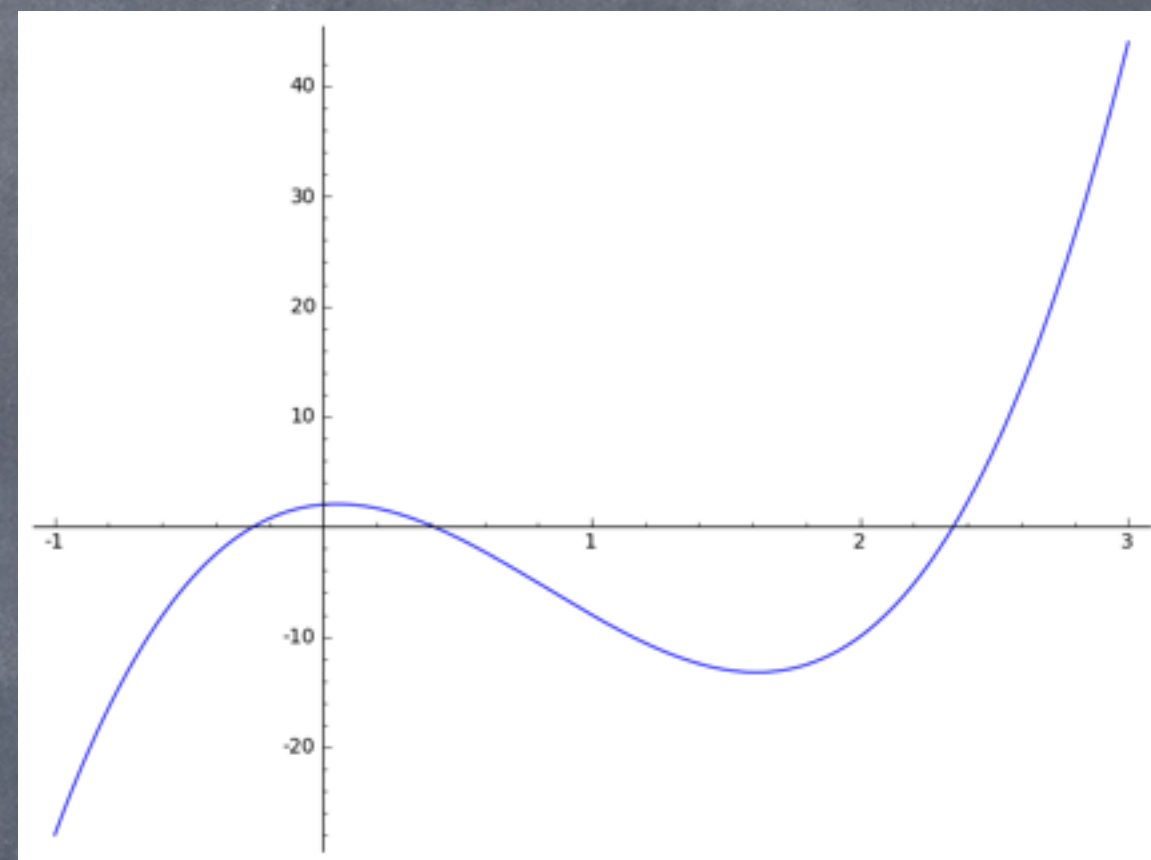
- [スツルムの定理]

$\sigma(\xi)$ を $p_0(\xi), p_1(\xi), \dots$ の
符号変化回数とすると、
区間 $(a, b]$ に含まれる実
根の個数は $\sigma(a) - \sigma(b)$ 個

- 根の限界で求めた区間
を、各区間に含まれる根
が1つになるまで分割

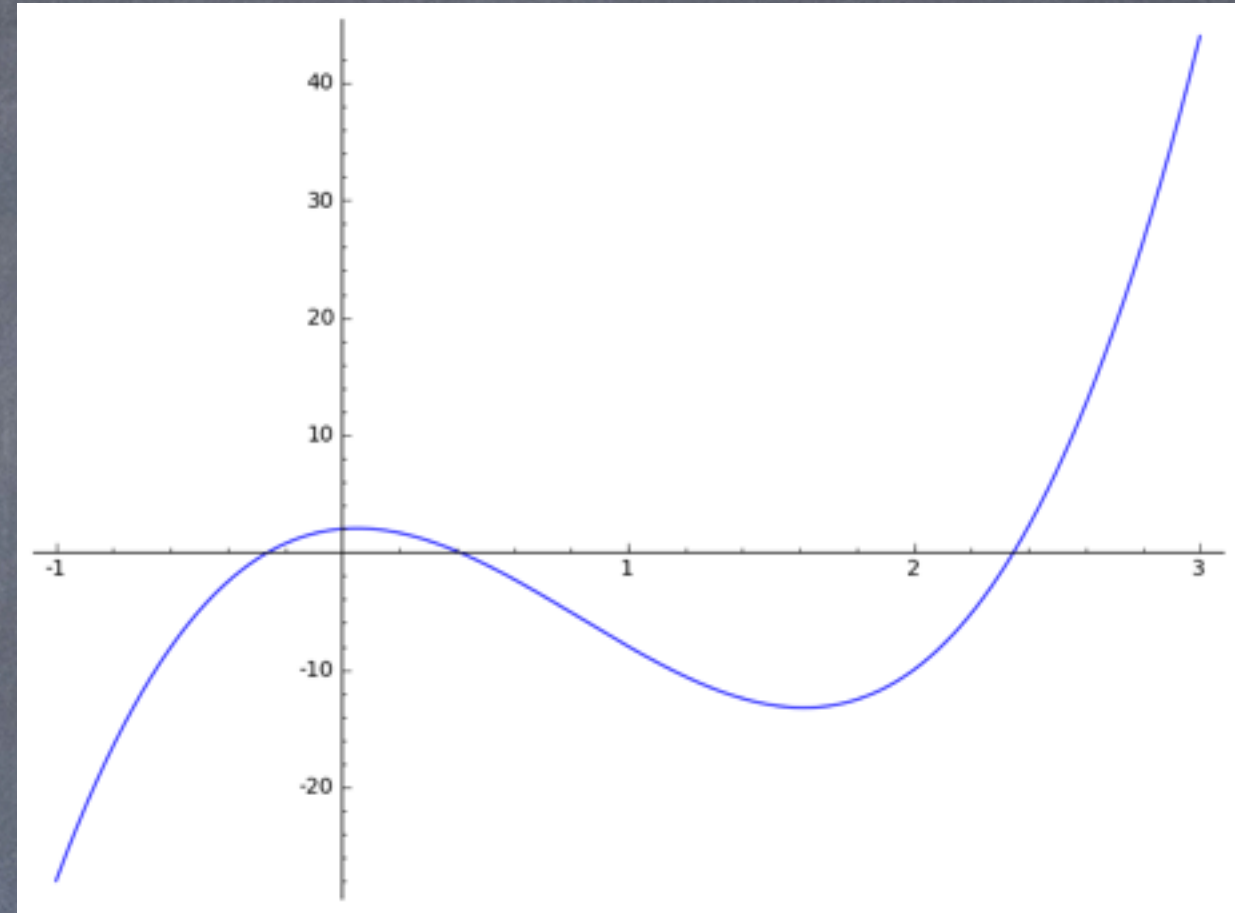
根の分離の例

- $p(x) = 4x^3 - 10x^2 + x + 1$ の場合
- 最初の区間は $[-3, 3]$
- $p(-3) = -200 \neq 0$ なので $(-3, 3]$
- スツルム鎖
 - $p_0 = 4x^3 - 10x^2 + x + 1$
 - $p_1 = 12x^2 - 20x + 1$
 - $p_2 = -(p_0 \bmod p_1) = (44/9)x - 23/18$
 - $p_3 = -(p_1 \bmod p_2) = 6597/1936$
- $\sigma(-3) = 3$... $(- \rightarrow + \rightarrow - \rightarrow +$ で3回の符号変化)
- $p_0(-3) = -200, p_1(-3) = 169, p_2(-3) = -287/18, p_3 = 6597/1936$
- 同様に $\sigma(3) = 0$... (0回の符号変化)
 - $p_0(3) = 22, p_1(3) = 49, p_2(3) = 241/18, p_3(3) = 6597/1936$
- $(-3, 3]$ には $\sigma(-3) - \sigma(3) = 3 - 0 = 3$ 個の根



根の分離の例

- $p(x) = 4x^3 - 10x^2 + x + 1$
- $(-3, 3]$ には 3 個の根
- $(-3, 3]$ を二等分して $(-3, 0], (0, 3]$
 - $(-3, 0]$ には 1 個の根 → OK
 - $(0, 3]$ には 2 個の根
- $(0, 3]$ を二等分して $(0, 3/2], (3/2, 3]$ に
 - $(0, 3/2]$ には 1 個の根 → OK
 - $(3/2, 3]$ には 1 個の根 → OK



目次

- ◉ 代数的実数とは
- ◉ 代数的実数の表現
- ◉ 根の求め方
- ◉ 代数的実数の演算
- ◉ 課題と今後

代数的実数の演算

- 近似値の計算
- 比較
- 四則演算など
- 代数的実数係数の多項式の根の計算

演算：近似値の計算

- スツルムの定理による区間の絞り込みで、任意有限精度で近似可能

- `approx`

`:: AReal → Rational → Rational`

- $\varepsilon > 0 \Rightarrow \text{abs} (\text{approx } a \ \varepsilon - a) \leq \varepsilon$

- `approxInterval`

`:: AReal → Rational → Interval Rational`

- $\varepsilon > 0 \Rightarrow \text{width} (\text{approxInterval } a \ \varepsilon) \leq \varepsilon$

演算: 比較

- Eq と Ord を実装
- $a = \text{AReal } p_1 \ I_1, b = \text{AReal } p_2 \ I_2$ とする
- $a = b$ の判定
 - $p_1 = p_2$ かつ $I_1 \cap I_2$ に根が含まれる
- $a < b$ や $a > b$ の判定
 - スツルムの定理で I_1 と I_2 を縮めていけば、区間が重ならなくなり、判定できる

演算: 四則演算など

- 適当な公理を満たす四則演算が定義できるので、
体になる
 - Num, Fractional, Real, RealFrac を実装
- 有理数との加算と乗算は簡単かつ高速
- 代数的実数同士の演算は面倒くさい

演算：有理数との演算

- 有理数との演算は変数変換と区間演算で容易に実現
- 代数的実数 $a = \text{AReal } p(x) \text{ } I$ と 有理数 b の加算の場合
 - $a + b = \text{AReal } p(x-b) (I+b)$
- 例
 - $\sqrt{2} + 3$
 $= (\text{AReal } (x^2-2) (0,2]) + 3$
 $= \text{AReal } ((x-3)^2-2) (0+3,2+3]$
 $= \text{AReal } (x^2 - 6x + 7) (3,5]$

演算: 有理数との演算

- 乗算の場合

- $a * b = \text{AReal } p(x/b) (I*b) \text{ if } b \neq 0$

- 例

- $$\begin{aligned} & \sqrt{2} * 3 \\ &= (\text{AReal } (x^2-2) (0,2]) * 3 \\ &= \text{AReal } ((x/3)^2-2) (0*3, 2*3] \\ &= \text{AReal } ((1/9)x^2 - 2x + 7) (0,6] \end{aligned}$$

代数的実数同士の加算

- $\alpha = \text{AReal } p_1(x) \text{ } I_1, \beta = \text{AReal } p_2(x) \text{ } I_2$
- $\alpha + \beta$ を根とする多項式をまず見つける
- すなわち
$$c_0 + c_1(\alpha+\beta) + \dots + c_{n-1}(\alpha+\beta)^{n-1} + c_n(\alpha+\beta)^n = 0$$
 を満たす n と $c_i \in \mathbb{Q}$ を見つけたい (ただし $c_i \neq 0$ for some j)
- $\{1, \alpha+\beta, (\alpha+\beta)^2, \dots, (\alpha+\beta)^n\}$ の線形従属性
- $\{1, x+y, (x+y)^2, \dots, (x+y)^n\} \subseteq \mathbb{Q}[x,y]/\{p_1(x), p_2(y)\}$ が線形従属になるまで、 n を増やししながら、試す

代数的実数同士の加算例

- $\sqrt{2} + \sqrt[3]{3}$

- $\{ 1, x+y, (x+y)^2, \dots, (x+y)^n \} \subseteq \mathbb{Q}[x,y]/\{x^2 - 2, y^3 - 3\}$

- $(x+y) = x+y$

- $(x+y)^2 = \underline{x^2} + 2xy + y^2 = 2 + 2xy + y^2$

- $(x+y)^3 = (x+y)(2 + 2xy + y^2)$
 $= 2x + 2\underline{x^2}y + xy^2 + 2y + 2xy^2 + \underline{y^3}$
 $= 2x + 4y + xy^2 + 2y + 2xy^2 + 3$
 $= 2x + 6y + 3xy^2 + 3$

- $(x+y)^4 = 8xy + 12y^2 + 12x + 3y + 4$

- ...

- 全て $\{ 1, x, y, xy, y^2, xy^2 \}$ の線形結合になるので、高々6次元までしかない

- n を増やしていけば、必ず線形従属に

代数的実数同士の加算例

- $c_0 + c_1(x+y) + \dots + c_{n-1}(x+y)^{n-1} + c_n(x+y)^n$ を
 $d_0 + d_1 x + d_2 y + d_3 xy + d_4 y^2 + d_5 xy^2$ という形に
変形して、連立方程式 $d_0 = d_1 = \dots = d_5 = 0$ を解く
- $n=3$ のとき
 - $c_0 + c_1(x+y) + c_2(x+y)^2 + c_3(x+y)^3$
 $= (c_0 + 2c_2 + c_3) + (c_1 + 2c_3)x + (c_1 + 6c_3)y + 2c_2xy + c_2y^2 + 3c_3xy^2$
 - 連立方程式 $(c_0 + 2c_2 + c_3) = (c_1 + 2c_3) = (c_1 + 6c_3) = 2c_2 = c_2 = 3c_3 = 0$
を解くと $c_0 = c_1 = c_2 = c_3 = 0$ なので線形独立

代数的実数同士を加算例

● $n=6$ のとき

$$\begin{aligned} & c_0 + c_1(x+y) + c_2(x+y)^2 + \dots + c_6(x+y)^6 \\ &= (c_0 + 2c_2 + 3c_3 + 4c_4 + 60c_5 + 17c_6) + (c_1 \\ &+ 2c_3 + 12c_4 + 4c_5 + 120c_6)x + \\ & \quad (c_1 + 6c_3 + 3c_4 + 20c_5 + 90c_6)y + (2c_2 + 8c_4 + 15c_5 + 24c_6)xy + \\ & \quad (c_2 + 12c_4 + 3c_5 + 60c_6)y^2 + (3c_3 + 20c_5 + 18c_6)xy^2 \end{aligned}$$

● 連立方程式 $(c_0 + 2c_2 + 3c_3 + 4c_4 + 60c_5 + 17c_6) = \dots =$
 $(3c_3 + 20c_5 + 18c_6) = 0$ を解くと

$$c_0 = c_6, \quad c_1 = -36c_6, \quad c_2 = 12c_6, \quad c_3 = -6c_6, \quad c_4 = -6c_6, \quad c_5 = 0, \quad c_6 = \text{free}$$

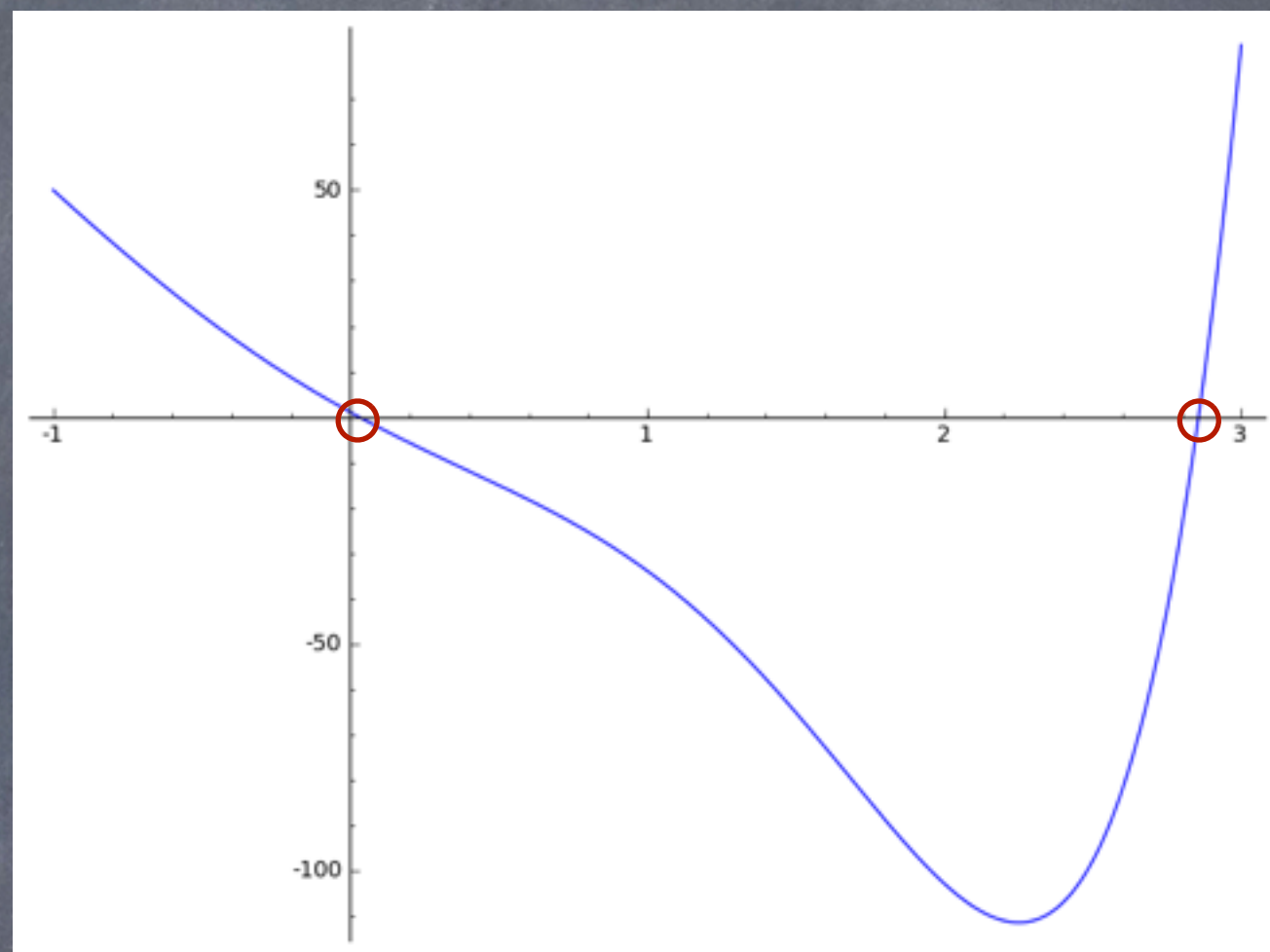
● とりあえず $c_6=1$ とおいて

$$q(z) = 1 - 36z + 12z^2 - 6z^3 - 6z^4 + z^6 = 0$$

が求める多項式

代数的実数同士の加算

- $a+b$ を根とする多項式を見つけたら、その多項式の根のうち、 $a+b$ 以外のものを区間演算で除外し、残ったものを返す



- $\sqrt{2} + \sqrt[3]{3}$ の場合

- 多項式 $q(z) = 1 - 36z + 12z^2 - 6z^3 - 6z^4 + z^6 = 0$

- $\sqrt{2} \in (0, 2]$, $\sqrt[3]{3} \in (3/4, 3/2]$ という区間で表現していたとすると、 $\sqrt{2} + \sqrt[3]{3} \in (3/4, 7/2]$ のはず。

- 区間を十分狭めれば小さい方の根は除外できる。

代数的実数のその他演算

- 乗算 $\alpha * \beta$

- $\{ 1, \alpha + \beta, (\alpha + \beta)^2, \dots, (\alpha + \beta)^n \}$ の代わりに $\{ 1, \alpha\beta, (\alpha\beta)^2, \dots, (\alpha\beta)^n \}$ を考えて加算の場合と同様にすれば良い

- 逆数 $1/\alpha$

- $p(1/x)$ の分母を払ったものを $q(x)$ とする

- 例: $1/\sqrt{2}$ の場合 $p(x) = x^2 - 2$

$$p(1/x) = (1/x)^2 - 2 \text{ の分母を払って } 1 - 2x^2$$

- その根のうちで適切なものを選ぶ

- 代数的実数係数の多項式の根の計算 (省略)

実装の補足

- 連立方程式の解き方
 - 掃き出し法とか実装するの面倒臭い
 - グレブナー基底使って計算
- Realクラスの `toRational :: a → Rational`
 - Haskellの設計者は有理数より詳細な実数型のことを考えてなかった？
 - 有理数でない場合にはエラーにすることに

目次

- ◉ 代数的実数とは
- ◉ 代数的実数の表現
- ◉ 根の求め方
- ◉ 代数的実数の演算
- ◉ 課題と今後

課題と今後

- 多項式の因数分解がボトルネックで非常に遅い
 - 現在 Kronecker's method という超素朴なアルゴリズムを使用
- 多項式の次数に対して指数的
 - $x^9 - 15x^6 - 87x^3 - 125$ の因数分解に対して、131072通りの候補で割ろうとしてた。そりゃ遅い..... orz
 - さらに代数的実数の加算や乗算では次数が最悪で積になる.....

課題と今後

- 多項式の因数分解がボトルネックで非常に遅い
 - $\Rightarrow$ より効率的な因数分解アルゴリズムを実装?
 - 有限体や p 進体上で因数分解して、その結果から有理数上での因数を構成するタイプのアルゴリズム
 - とりあえず、有限体上の因数分解のBerlekampアルゴリズムは実装
 - とりあえず、素体は実装 (finite-fieldパッケージ)
 - $\Rightarrow$ 完全な因数分解を遅延?
 - 完全に因数分解しなくても、重複因子がなくなるまで因数分解しておけば、ほとんどの演算は可能

課題と今後

- コードと依存関係を整理して Hackage に公開したい
 - 自前の多項式ライブラリもパッケージ化?
 - それとも既存の多項式ライブラリに乗り換える?
- 実数に限らない代数的数も取り扱いたい
 - 根の分離が課題? Graeffe's Method ?
 - 単に代数的実数の組で扱えば良い?

後半：実閉体の決定可能性

実閉体とは

- 以下の条件を満たす体
 - 順序体
 - $a \leq b \Rightarrow a + c \leq b + c$
 - $0 \leq a \wedge 0 \leq b \Rightarrow 0 \leq ab$
 - 奇数次の多項式は根を持つ
 - 正の元は平方根を持つ
- 具体例: 実数, 代数的実数, 計算可能実数, 超実数
- 順序体の言語に関して、実数の場合と真偽が一致

実閉体の理論の決定可能性

- ここで考える論理式

- 有理数係数の多項式に関する等式・不等式を原子論理式とする
一階述語論理の論理式

- 例: $\forall a, b, u. (a \leq b \wedge a^2 \leq u \leq b^2) \rightarrow \exists c. a \leq c \leq b \wedge u = c^2$

- 決定可能性・完全性

- 実数論: 論理式の真偽を決定可能、どのモデルでも同じ真偽
- 自然数論: $\forall$ を決定不能、モデルによって真偽が異なる

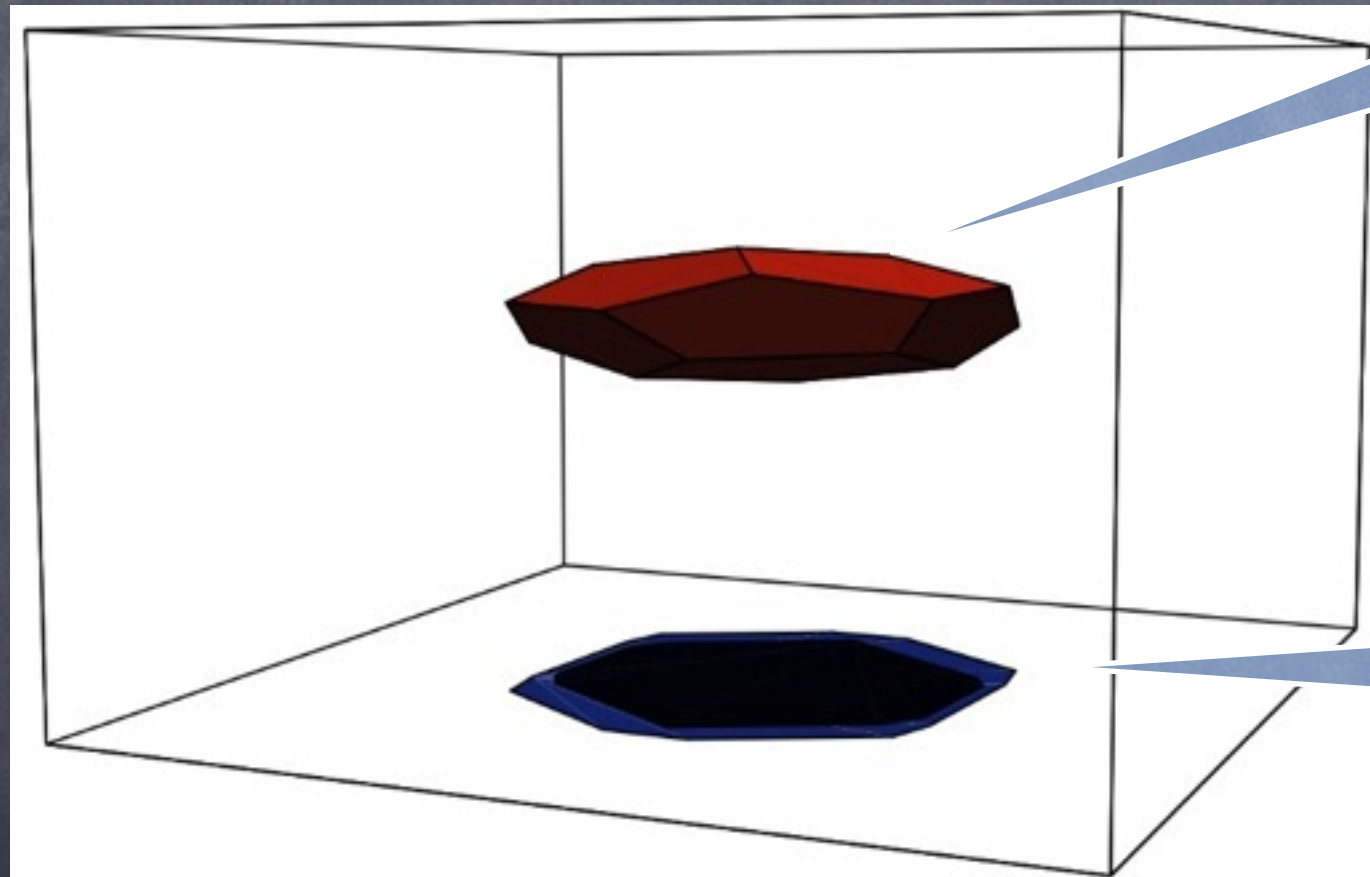
- Tarskiが1951年に量子化除去で証明

量子化子除去

- $P(x,y)$ を量子化子を含まない論理式とする
- $\exists y P(x,y)$ を、それと同値で量子化子を含まない論理式 $Q(x)$ へと変換
 - 例: $(\exists x. ax^2 + bx + c = 0) \Leftrightarrow (b^2 - 4ac \geq 0)$
- 変換を繰り返し適用して、すべての量子化子を除去
- 変数をすべて消去すれば真偽は自明

量子化と射影

$$\exists z. P(x,y,z) \Leftrightarrow Q(x,y)$$



$$\{(x,y,z) \mid P(x,y,z)\}$$

$$\{(x,y) \mid \exists z. P(x,y,z)\} \\ = \{(x,y) \mid Q(x,y)\}$$

The Omega Test: a fast and practical integer programming algorithm for dependence analysis

他の量子子除去 アルゴリズム

- ProofSummit2012 で少し紹介
 - Fourier-Motzkin (実数上の線形算術)
 - Cooper, OmegaTest (自然数・整数の線形算術)
- 余談
 - そういえば、今回の話はProofSummit2013で紹介しようと思ってたネタだったのをすっかり忘れてた.....

Cylindrical Algebraic Decomposition (CAD)

- Tarskiが用いた量化子除去アルゴリズムは
すごく計算量が大きい
 - non-elementary (多重指数で抑えられない)
- Collinsが1975により効率的な手法
Cylindrical Algebraic Decomposition (CAD)を提案
 - double-exponential
- 今回はそれをベースに説明

扱う論理式の形

- $\varphi(y_1, \dots, y_m) = \exists x. \bigwedge_{i=1}^n p_i(x, y_1, \dots, y_m) \bowtie_i 0$
 - $p_i(x, y_1, \dots, y_m) \in Q[y_1, \dots, y_m][x]$
 - $\bowtie_i \in \{=, <, >\}$
- という形の論理式の量子子除去を扱う
- 他の場合はこれに帰着

- Something you should know about: Quantifier Elimination (Part II)

- by Arnab Bhattacharyya

- の説明が秀逸だったので、その解説

- 以下の順で一步一步説明

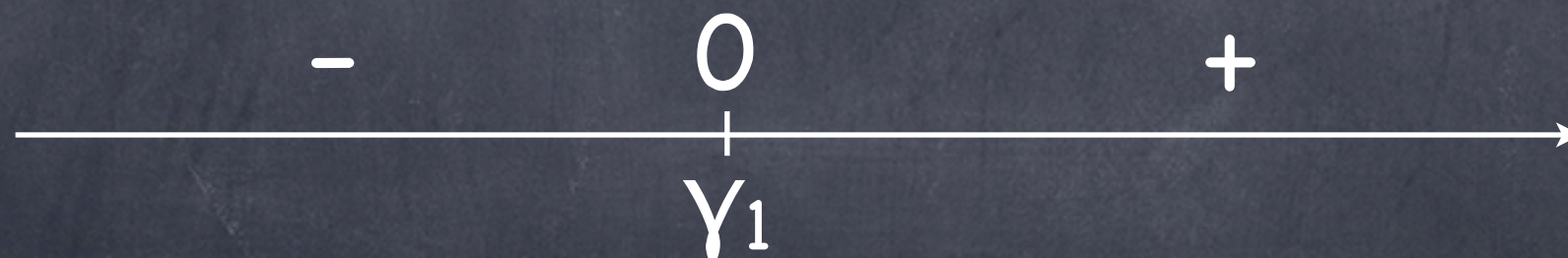
- 1変数かつ線形の場合

- 1変数かつ非線形の場合

- 多変数の場合 (自由変数がある場合)

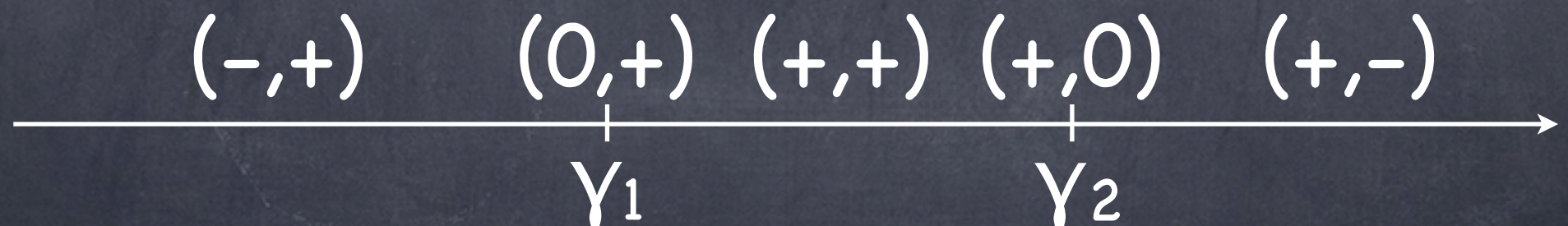
1変数かつ線形の場合

- $\varphi = \exists x. (x+1 > 0) \wedge (-2x+3 > 0) \wedge (x > 0)$
- $p_1(x) = x+1$ の根を γ_1 と置く
- p_1 は γ_1 で 0, 根の左では常に負、右では常に正



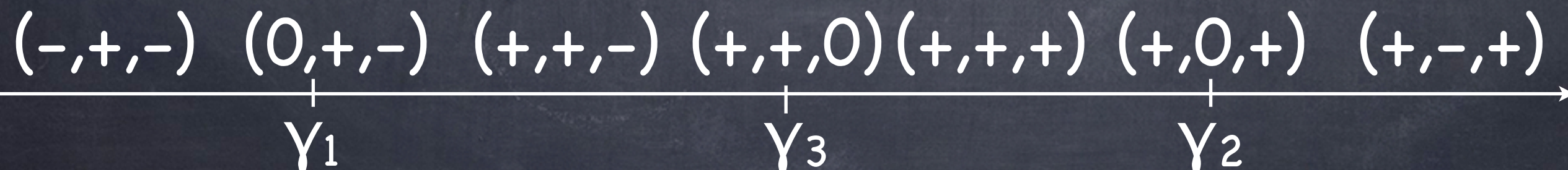
1変数かつ線形の場合

- $\varphi = \exists x. (x+1 > 0) \wedge (-2x+3 > 0) \wedge (x > 0)$
- 次に $p_2(x) = -2x+3$ の根 γ_2 を考える
- p_2 は γ_2 の左では常に正、右では常に負
- $p_2(x) = -2x+3 = -2p_1(x) + 5$ より $p_2(\gamma_1)=5 > 0$
- よって γ_1 は γ_2 の左



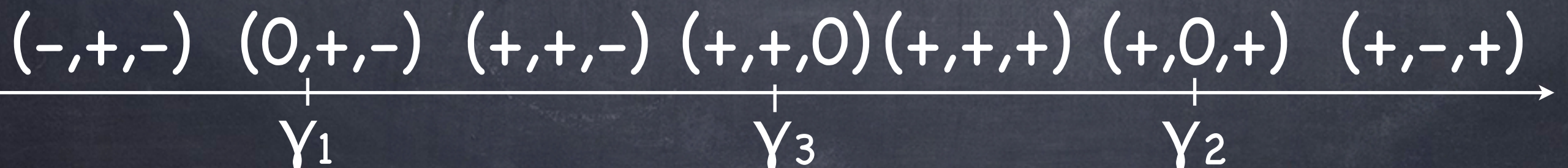
1変数かつ線形の場合

- $\varphi = \exists x. (x+1 > 0) \wedge (-2x+3 > 0) \wedge (x > 0)$
- 次に $p_3(x) = x$ の根 γ_3 を考える
- p_3 は γ_3 の左では常に負、右では常に正
- $p_3(x) = x = p_1(x) - 1$ より $p_3(\gamma_1) = -1 < 0$
よって γ_1 は γ_3 の左
- $p_3(x) = x = (-1/2)p_2(x) + 5/2$ より $p_3(\gamma_2) = 5/2 > 0$
よって γ_2 は γ_3 の右



1変数かつ線形の場合

- $\varphi = \exists x. (x+1 > 0) \wedge (-2x+3 > 0) \wedge (x > 0)$
- 区間 (γ_2, γ_3) でこの量子子の本体は真になるので、
 $\varphi \Leftrightarrow \top$
- γ_2, γ_3 を具体的に求めて、 (γ_3, γ_2) から値を選べば、
量子子の本体を満たす値も具体的に得られる



1変数かつ線形の場合

まとめ

- 符号表 (sign configuration diagram) を生成
 - 区間 $(-\infty, \infty)$ から始めて、多項式の根で区間を分割していく
 - $p(a)$ と $p(b)$ の符号が異なれば、区間 (a, b) に p の根
 - q の根 γ での p の符号は $p \bmod q$ の符号に等しい
- 最初の論理式に合致するような符号の組み合わせがあれば真

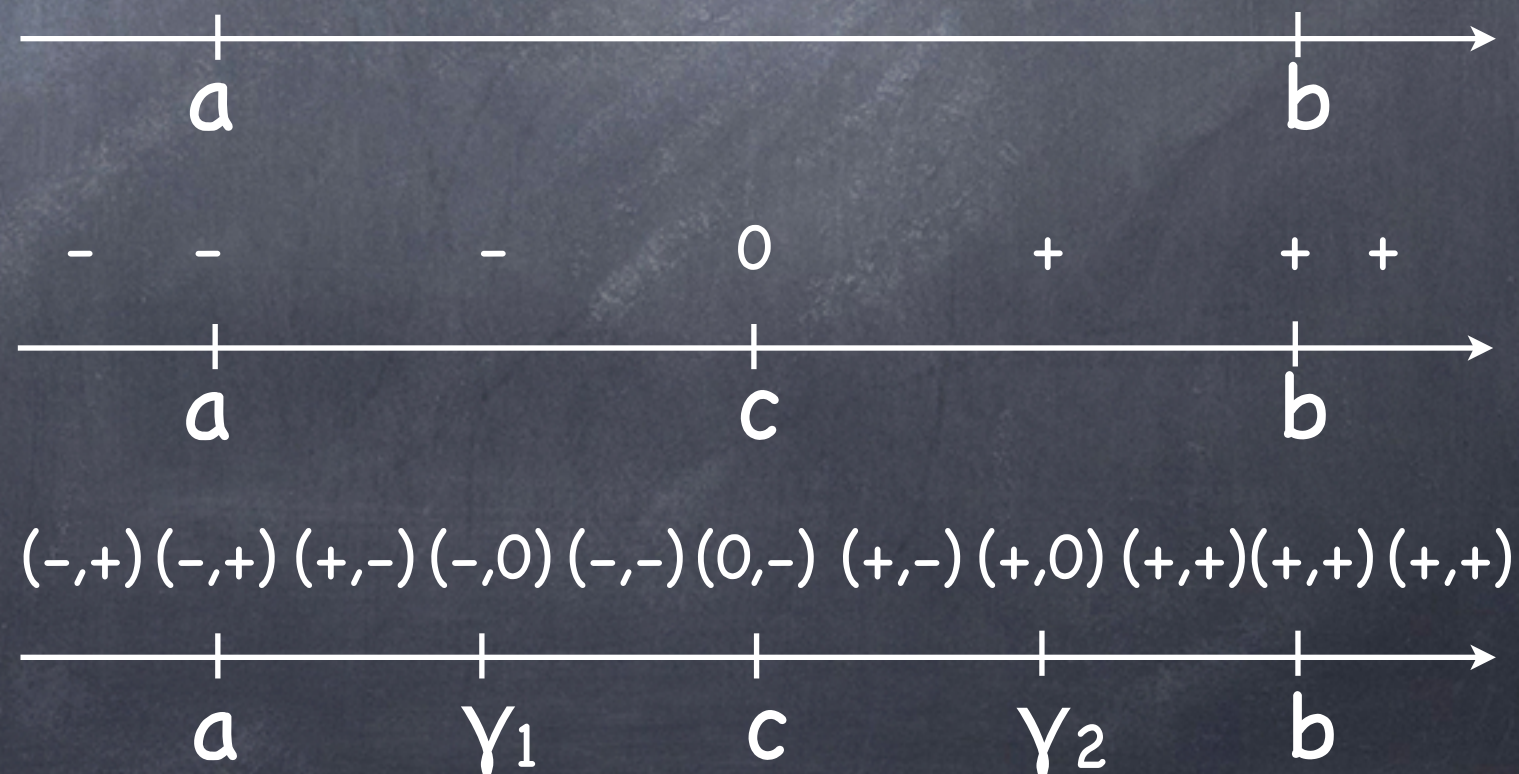
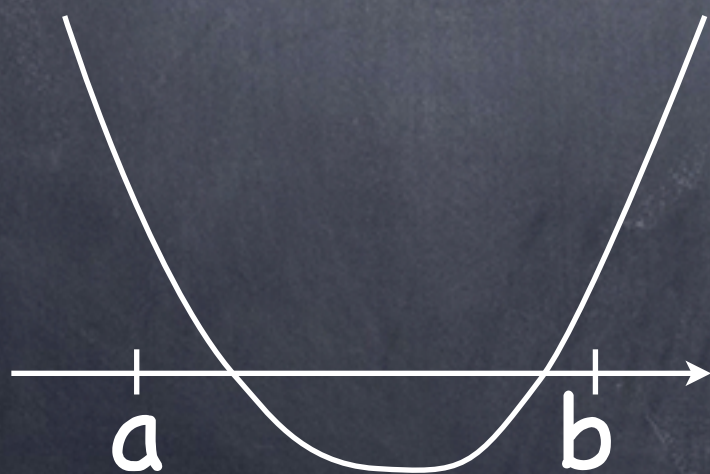
1変数かつ非線形の場合

- 線形の場合 (a,b) に根があることと、
 $p(a)$ と $p(b)$ の符号が異なることが同値だった
- 非線形の場合には $p(a)$ と $p(b)$ の符号が等しくても、 (a,b) に根があることがある



1変数かつ非線形の場合

- 非線形の場合には $p(a)$ と $p(b)$ の符号が等しくても、 (a,b) に根があることがある
- その場合 $p'(x)$ の符号が $c \in (a,b)$ で変化
- なので、 $p'(x)$ の符号 で先に区間分割



1変数かつ非線形の場合

- q の根 γ での p の符号は $p \bmod q$ の符号に等しい
- 線形の場合には、 $p \bmod q$ は常に定数だった ($\deg(q) > 0$ なら)
- 非線形の場合 $p \bmod q$ は一般の多項式
- $\Rightarrow p \bmod q$ の符号を先に求めておく
($p \bmod q$ で先に区間分割しておく)

1変数かつ非線形の場合

- $p(x)$ での区間分割の前に必要なこと
 - $p'(x)$ での区間分割
 - $p \bmod q$ ($\deg(q) < \deg(p)$) での区間分割
- 多項式集合 P から微分と $\bmod$ で生成される多項式の集合を考えて、次数の低いものから順に区間分割していく

多変数の場合

- $\varphi(y_1, \dots, y_m) = \exists x. \bigwedge_{i=1}^n p_i(x, y_1, \dots, y_m) \bowtie_i 0$

- $p_i(x, y_1, \dots, y_m) \in Q[y_1, \dots, y_m][x]$

- φ に自由変数 $y_1, \dots, y_m$ が含まれる場合

- 説明面倒臭いので省略したい

が、さすがにこれを省略するわけにはいかない.....

仕方ないのでちょっとだけ説明します

多変数の場合

- 多変数の場合の多項式の剰余をどうする?
 - $p(x,y) = p_D(y)x^D + p_{D-1}(y)x^{D-1} + \dots + p_0(y)$
 - $q(x,y) = q_E(y)x^E + q_{E-1}(y)x^{E-1} + \dots + q_0(y)$
- $p \bmod q$ は p_d を q_e で割り切れるとは限らない orz
- $\Rightarrow$ 符号を計算するのに都合の良い代わりの演算
 - $p \bmod q = (q_E(y)^{D-E+1} p = q s + r \text{ である一意な } r)$
 - $(q \text{ の根 } \gamma \text{ での } p \text{ の符号}) =$
 $(p \bmod q \text{ の符号}) / (q_E(\gamma)^{D-E+1} \text{ の符号})$

多変数の場合

- 多変数の場合の多項式の剰余をどうする?
 - $p(x,y) = p_D(y)x^D + p_{D-1}(y)x^{D-1} + \dots + p_0(y)$
 - $q(x,y) = q_E(y)x^E + q_{E-1}(y)x^{E-1} + \dots + q_0(y)$
- 符号を計算するのに都合の良い代わりの演算
 - $p \bmod q = (q_E(y)^{D-E+1} p = q s + r \text{ である一意な } r)$
 - $(q \text{ の根 } \gamma \text{ での } p \text{ の符号}) =$
 $(p \bmod q \text{ の符号}) / (q_E(y)^{D-E+1} \text{ の符号})$
- $q_E(y)$ の符号も分かんないじゃん!!
 - 符号 $\{0,+,-\}$ で場合分けして、結果の論理式の“or”を取る
 - $q_E(y)=0$ だと困るので、「仮定した符号」が非ゼロの最高次のものを $q_E(y)$ とする

実装しました！

実験: $\exists x. a^2x+bx+c = 0$

$$(a = 0 \wedge 2^*a = 0 \wedge b = 0 \wedge c = 0) \vee$$

$$(a = 0 \wedge 2^*a = 0 \wedge b < 0) \vee$$

$$(a = 0 \wedge 2^*a = 0 \wedge b > 0) \vee$$

$$(a = 0 \wedge 2^*a < 0 \wedge 2^*a^*c - b^2 < 0 \wedge b < 0) \vee$$

$$(a = 0 \wedge 2^*a < 0 \wedge 2^*a^*c - b^2 = 0 \wedge b < 0) \vee$$

$$(a = 0 \wedge 2^*a < 0 \wedge 2^*a^*c - b^2 > 0 \wedge b < 0) \vee$$

$$(a = 0 \wedge 2^*a < 0 \wedge 2^*a^*c - b^2 < 0 \wedge b > 0) \vee$$

$$(a = 0 \wedge 2^*a < 0 \wedge 2^*a^*c - b^2 = 0 \wedge b > 0) \vee$$

$$(a = 0 \wedge 2^*a < 0 \wedge 2^*a^*c - b^2 > 0 \wedge b > 0) \vee$$

$$(a = 0 \wedge 2^*a > 0 \wedge 2^*a^*c - b^2 < 0 \wedge b < 0) \vee$$

$$(a = 0 \wedge 2^*a > 0 \wedge 2^*a^*c - b^2 = 0 \wedge b < 0) \vee$$

$$(a = 0 \wedge 2^*a > 0 \wedge 2^*a^*c - b^2 > 0 \wedge b < 0) \vee$$

～右に続く～

～左からの続き～

$$(a = 0 \wedge 2^*a > 0 \wedge 2^*a^*c - b^2 < 0 \wedge b > 0) \vee$$

$$(a = 0 \wedge 2^*a > 0 \wedge 2^*a^*c - b^2 = 0 \wedge b > 0) \vee$$

$$(a = 0 \wedge 2^*a > 0 \wedge 2^*a^*c - b^2 > 0 \wedge b > 0) \vee$$

$$(a < 0 \wedge 2^*a < 0 \wedge 4^*a^2c - a^*b^2 = 0) \vee$$

$$(a < 0 \wedge 2^*a < 0 \wedge 4^*a^2c - a^*b^2 > 0) \vee$$

$$(a > 0 \wedge 2^*a < 0 \wedge 4^*a^2c - a^*b^2 < 0) \vee$$

$$(a > 0 \wedge 2^*a < 0 \wedge 4^*a^2c - a^*b^2 = 0) \vee$$

$$(a < 0 \wedge 2^*a > 0 \wedge 4^*a^2c - a^*b^2 = 0) \vee$$

$$(a < 0 \wedge 2^*a > 0 \wedge 4^*a^2c - a^*b^2 > 0) \vee$$

$$(a > 0 \wedge 2^*a > 0 \wedge 4^*a^2c - a^*b^2 < 0) \vee$$

$$(a > 0 \wedge 2^*a > 0 \wedge 4^*a^2c - a^*b^2 = 0)$$

実験: $\exists x. a^2x+bx+c = 0$

矛盾するものを手で消してみた

$$(a = 0 \wedge 2*a = 0 \wedge b = 0 \wedge c = 0) \vee$$

$$(a = 0 \wedge 2*a = 0 \wedge b < 0) \vee$$

$$(a = 0 \wedge 2*a = 0 \wedge b > 0) \vee$$

$$(a < 0 \wedge 2*a < 0 \wedge 4*a^2*c - a*b^2 = 0) \vee$$

$$(a < 0 \wedge 2*a < 0 \wedge 4*a^2*c - a*b^2 > 0) \vee$$

$$(a > 0 \wedge 2*a > 0 \wedge 4*a^2*c - a*b^2 < 0) \vee$$

$$(a > 0 \wedge 2*a > 0 \wedge 4*a^2*c - a*b^2 = 0)$$

一般の論理式の場合

- 任意の論理式は以下に変換可能 (φ は量子子を含まない)

$$Q_1 x_1 \ Q_2 x_2 \ \dots \ Q_n x_n \ \varphi \ (Q_i \in \{\forall, \exists\})$$

- これに対して内側の量子子から順に除去する

- $Q_n = \exists$ なら

$$\exists x_n \ \varphi = \exists x_n \ (\bigvee_j \bigwedge_k \varphi_{j,k}) = \bigvee_j (\exists x_n. \bigwedge_k \varphi_{j,k})$$

と変形して、選言肢のそれぞれで量子子除去

- $Q_n = \forall$ なら $\forall x_n \ \varphi = \neg(\exists x_n \neg \varphi)$ と変形して

$\exists x_n$ の量子子除去と同様にする

まとめ

- 実閉体の理論は決定可能
- 量子子除去の副産物として得られる符号表から実際に制約を満たす値を計算可能
 - 結果は代数的実数で表現
- 量子子除去結果は場合分け爆発するので、論理式の単純化処理が必要
 - 最近では BDD/ZDD で単純化みたいな話も
- その他最適化の余地は色々

参考文献

- The beginnings of Galois theory
 - by Timothy Gowers
 - <http://www.dpmms.cam.ac.uk/~wtg10/galois.html>
- Something you should know about: Quantifier Elimination
 - by Arnab Bhattacharyya
 - <http://cstheory.blogoverflow.com/2011/11/something-you-should-know-about-quantifier-elimination-part-i/>
 - <http://cstheory.blogoverflow.com/2012/02/something-you-should-know-about-quantifier-elimination-part-ii/>
- Quantifier Elimination following Muchnik
 - by Christian MICHAUX and Adem OZTURK
 - <https://math.umons.ac.be/preprints/src/Ozturk020411.pdf>