

CS1 (10): テスティング落穂拾い

脇田建

2015.12.22

Mandelbrotのテストの続き

前回のテスト

現象の分析	テストの記述
最初: (R1)	S0 should be (R1)
拡大 → (R2 , R1)	S1 = Drag (S0) should be (R2 , R1)
拡大 → (R3 , R2, R1)	S2 = Drag (S1) should be (R3 , R2, R1)
戻る → (R3, R2 , R1)	S3 = Bwd (S2) should be (R3, R2 , R1)
戻る → (R3, R2, R1)	S4 = Bwd (S3) should be (R3, R2, R1)
戻る → (R3, R2, R1)	S5 = Bwd (S4) should be (R3, R2, R1)
進む → (R3, R2 , R1)	S6 = Fwd (S5) should be (R3, R2 , R1)
進む → (R3 , R2, R1)	S7 = Fwd (S6) should be (R3 , R2, R1)
進む → (R3 , R2, R1)	S8 = Fwd (S7) should be (R3 , R2, R1)

test/mandelbrot4.scala

$S3 = \text{Bwd}(S2)$ should be $(R3, \text{R2}, R1)$

- ❖ $T(\text{"DDBDBBB"}, \text{"X_X_X_X_X_X_X"})$
- ❖ 拡大, 拡大, Bwd, 拡大, Bwd, ... した各段階を一気にテスト.

森先生とTAの講評

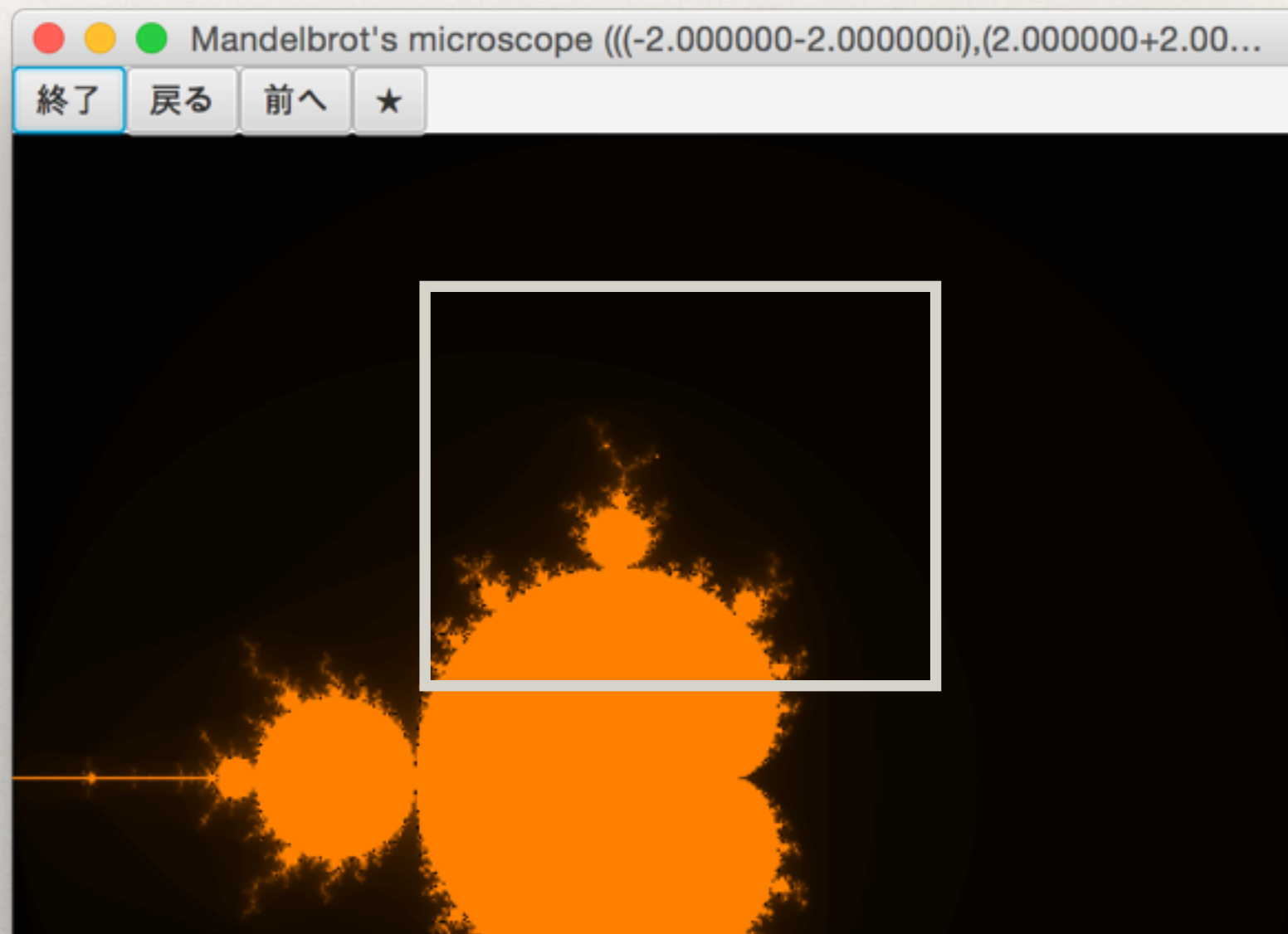
- ❖ 履歴情報の領域の変化は確かにテストされています
- ❖ でも、履歴情報に含まれている座標についてはテストしていないんじゃないですか？

たとえば

- ❖ ドラッグで表示画面の内側を選択しているのに，実際にはその外側が表示されてはいないだろうか？
- ❖ （実は前回のコードにはこのバグがあった．後述）
- ❖ ドラッグで表示された画面情報以外の履歴情報は本当に破壊されていないだろうか？

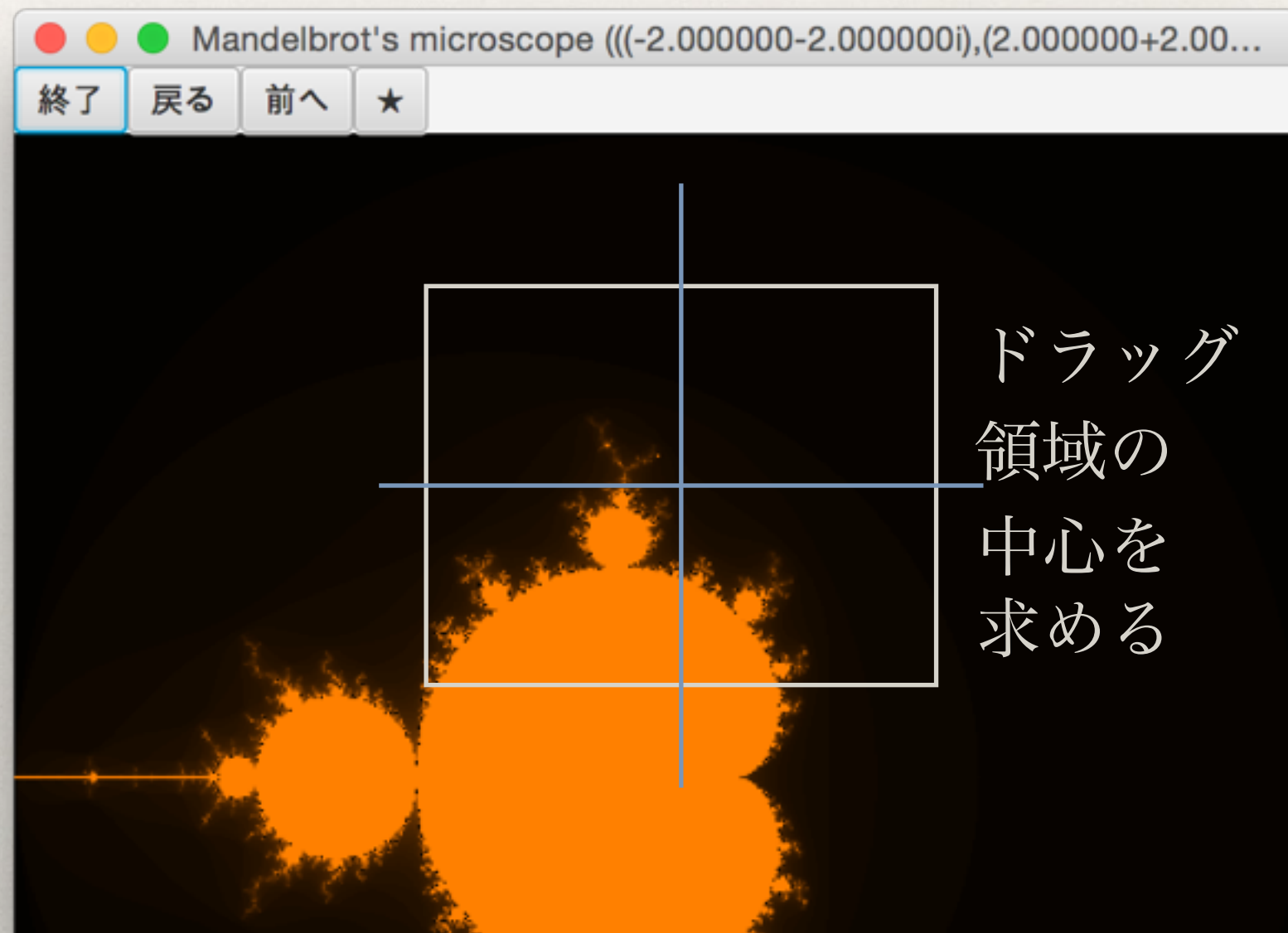
脇田のバグつきアルゴリズム

- ❖ ドラッグで表示画面の内側を選択しているのに，実際にはその外側が表示されてはいないだろうか？



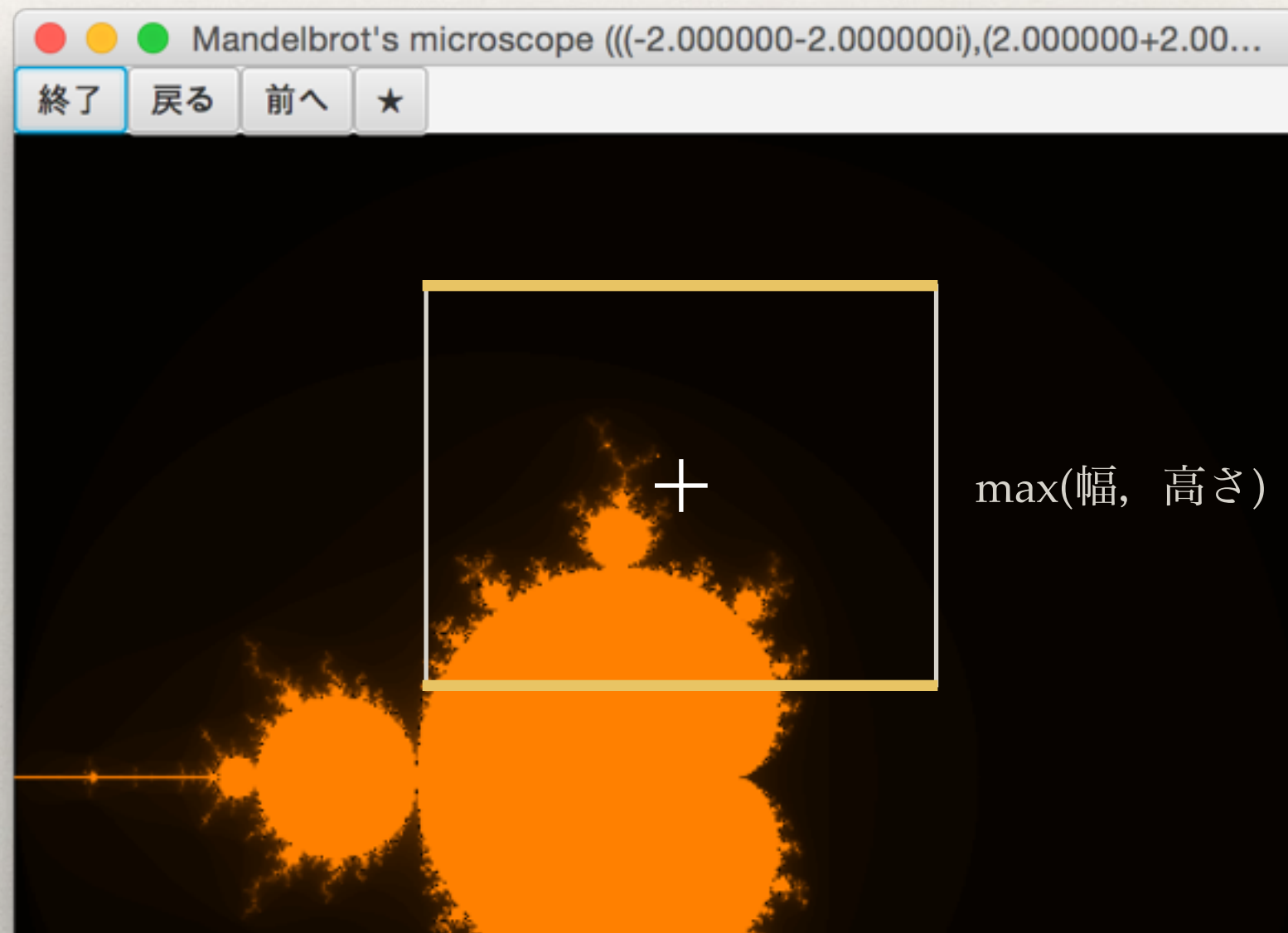
脇田のバグつきアルゴリズム

- ❖ ドラッグで表示画面の内側を選択しているのに，実際にはその外側が表示されてはいないだろうか？



脇田のバグつきアルゴリズム

- ❖ ドラッグで表示画面の内側を選択しているのに，実際にはその外側が表示されてはいないだろうか？



脇田のバグつきアルゴリズム

- ❖ ドラッグで表示画面の内側を選択しているのに，実際にはその外側が表示されてはいないだろうか？



脇田のバグつきコード

```
46 def subRegion(h: History, p1: Complex, p2: Complex) = {  
47     val size = max(abs(p2.re - p1.re), abs(p2.im - p1.im))  
48     val c1 = new Complex((p1.re + p2.re - size) / 2, (p1.im + p2.im - size) / 2)  
49     val c2 = new Complex((p1.re + p2.re + size) / 2, (p1.im + p2.im + size) / 2)  
50     ((c1, c2) :: dropToCurrent(h), 0)  
51 }
```

テストケース

```
// 履歴の底は初期画面であること
h2(h2.length - 1) == currentRegion(history) should be (true)

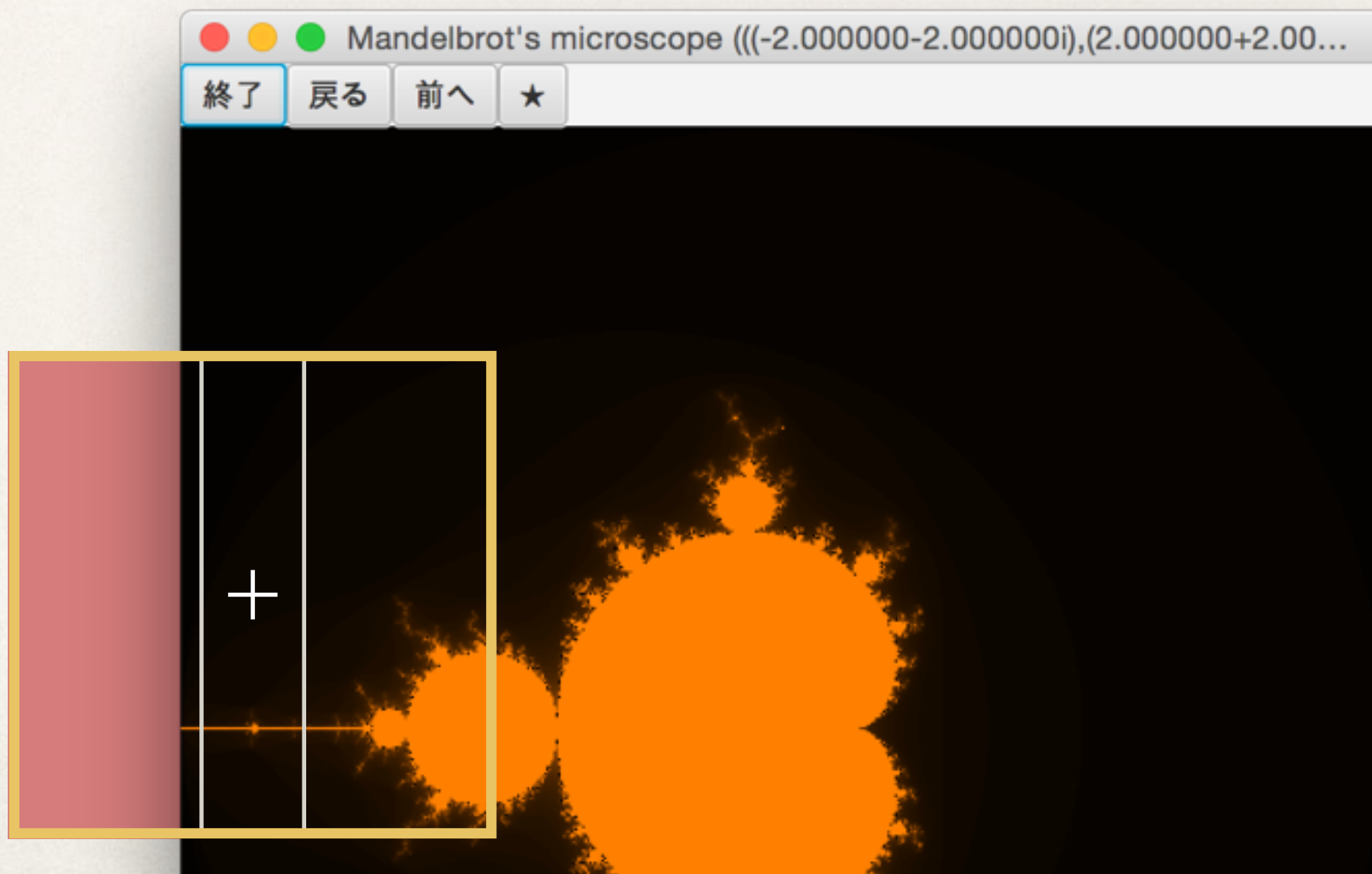
// 履歴情報に含まれる表示範囲について包含関係が成立すること、すなわち
// 履歴情報: [ Rn, ..., R2, R1 ] のとき Rn \subset ... \subset R2 \subset R1 (= (-2-2i, 2+2i))

for (i <- 0 to h2.length - 2) {
  (h2(i), h2(i+1)) match {
    case ((c21, c22), (c11, c12)) => {
      List(c11.re, c21.re, c22.re, c12.re) shouldBe sorted
      List(c11.im, c21.im, c22.im, c12.im) shouldBe sorted
    }
  }
}
```

❖ なんと、すべてのテストケースが失敗

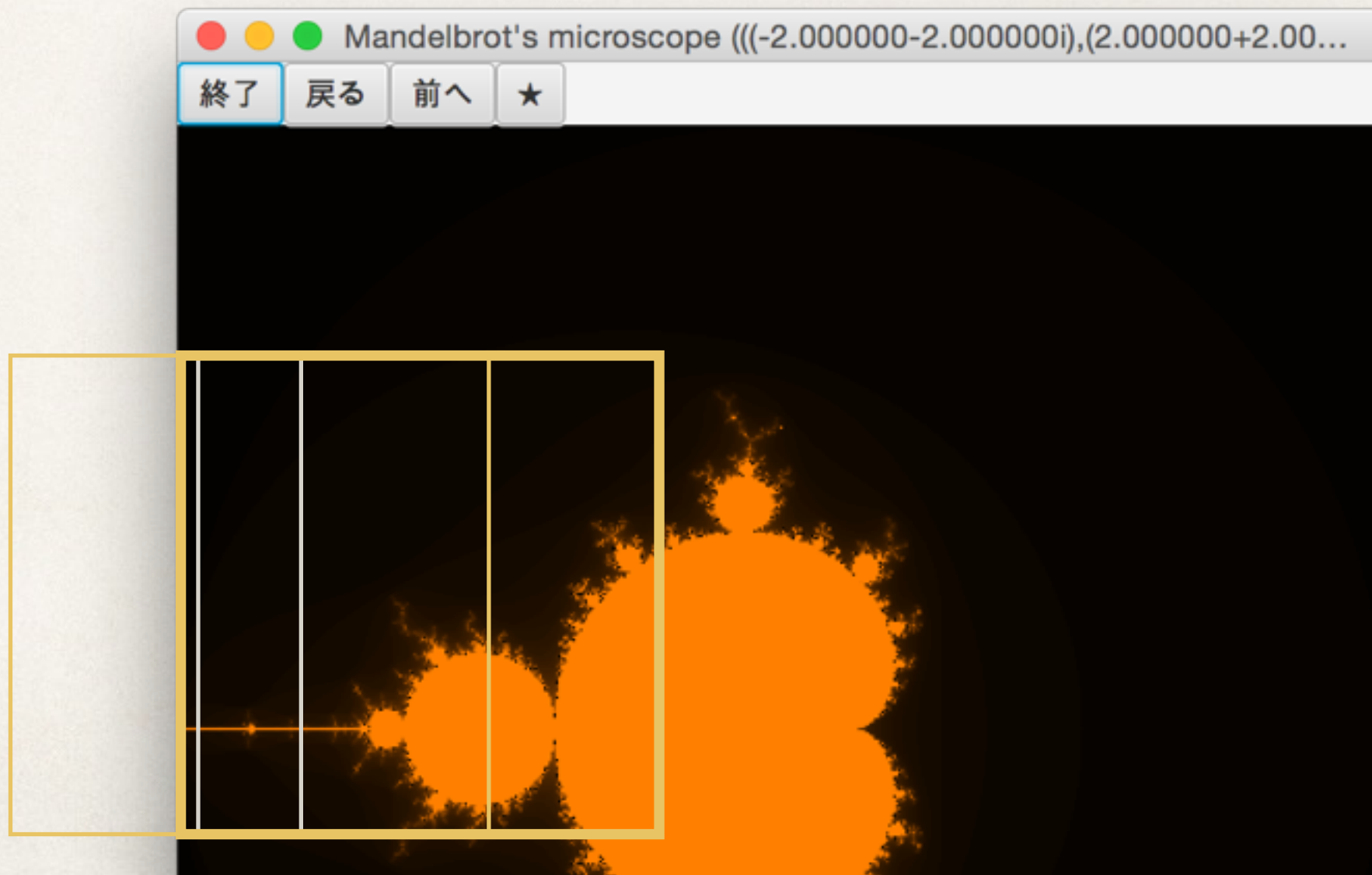
脇田のバグの種あかし

- ❖ 表示画面のすぐ内側の細長い領域が選択されると、表示画面からはみだした領域が選択される。



アルゴリズムの修正

- ❖ はみだしそうになったら，元の表示画面内に押し込む.



テストケースに併せてバグの修正

```
def subRegion(h: History, p1: Complex, p2: Complex) = {  
  currentRegion(h) match { case (c1, c2) =>  
    val size = max(abs(p2.re - p1.re), abs(p2.im - p1.im))  
    val _re1 = (p1.re + p2.re - size) / 2  
    val _im1 = (p1.im + p2.im - size) / 2  
  
    val re1 = min(max(c1.re, _re1), c2.re - size)  
    val im1 = min(max(c1.im, _im1), c2.im - size)  
  
    ((new Complex(re1, im1), new Complex(re1 + size, im1 + size)) ::  
     dropToCurrent(h), 0)  
  }  
}
```

もうひとつの検査項目

- ❖ ドラッグで表示された画面情報以外の履歴情報は本当に破壊されていないだろうか？

新たな検査仕様

❖ 履歴情報のうち現在表示されてもの x or X

❖ X（大文字）直前の操作で履歴に追加された新規データ

❖ x, _ : 直前の操作で変化しない

❖ D(rag)

❖ F(orward)

❖ B(ackward)

```
T("B", "x")  
T("F", "x")
```

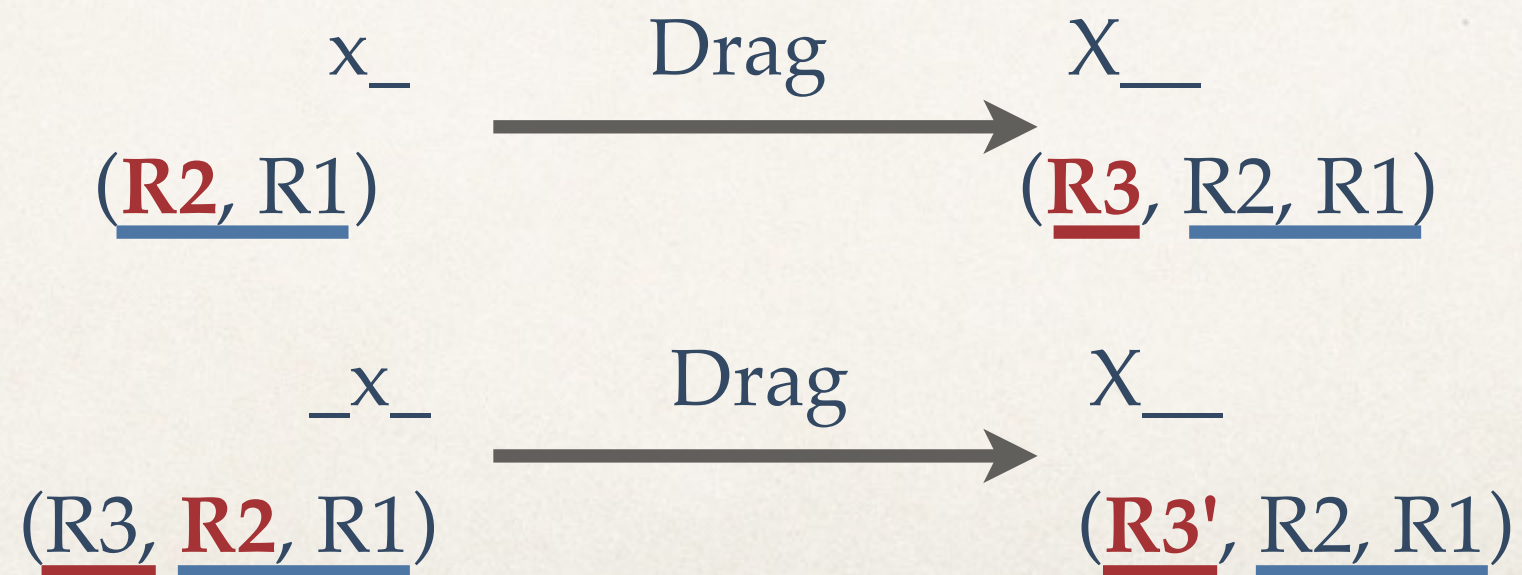
```
T("DF", "X_,x_")  
T("DBB", "X_,_x,_x")
```

```
T("DDF", "X_,X_,x_")  
T("DDBBB", "X_,X_,_x_,_x,_x")
```

```
T("DDBFF", "X_,X_,_x_,x_,x_")  
T("DDBBB", "X_,X_,_x_,_x,_x")
```

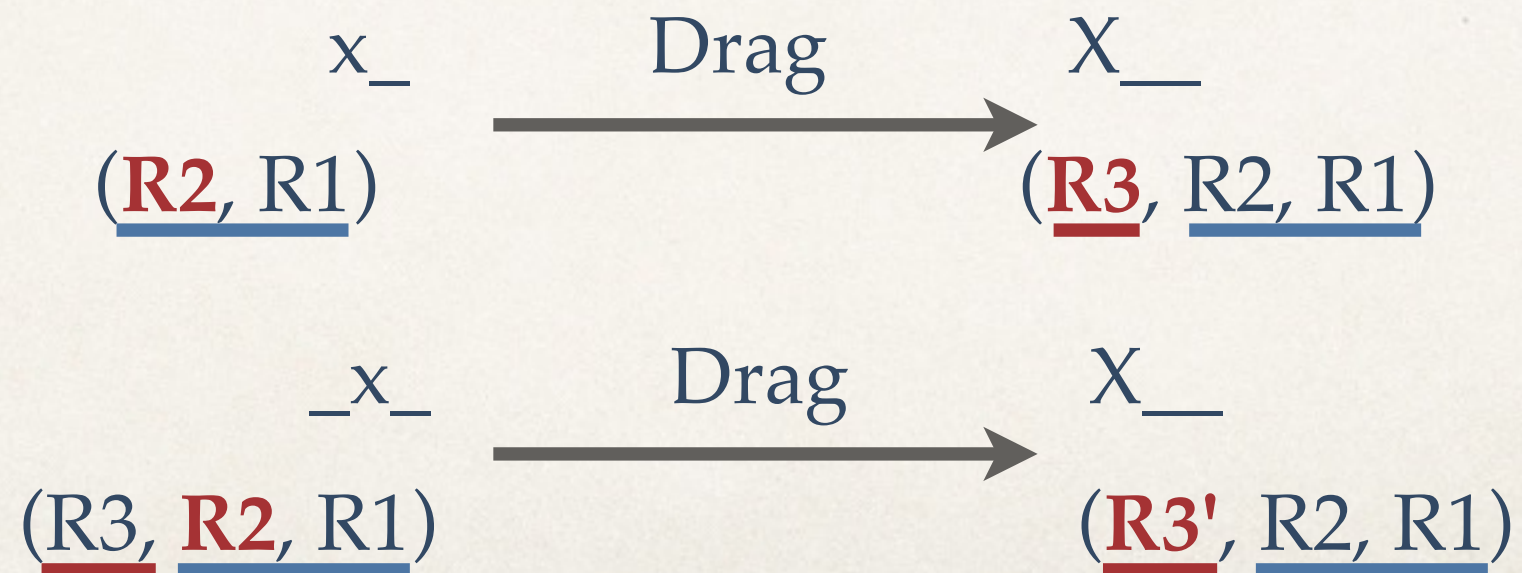
```
T("DDBDF", "X_,X_,_x_,X_,x_")  
T("DDBDBBB", "X_,X_,_x_,X_,_x_,_x,_x")
```

```
// 履歴情報のうちドラッグされたばかりの領域(X)のみが変更されていること
val scenes = answer.split("")
val len1 = h1.length
val len2 = h2.length
for (s <- scenes.indices) {
  val i = len2 - 1 - s
  if (i < len1 - 1) {
    val unchanged = (h2(s) == h1(len1 - 1 - i))
    unchanged should be (scenes(s) != 'X')
  }
}
```




```
// 履歴情報のうちドラッグされたばかりの領域 (X)のみが変更されていること
val scenes = answer.split("")
val len1 = h1.length
val len2 = h2.length
for (s <- scenes.indices) {
  val i = len2 - 1 - s
  if (i < len1 - 1) {
    val unchanged = (h2(s) == h1(len1 - 1 - i))
    unchanged should be (scenes(s) != 'X')
  }
}
```

オブジェクトの同一性判定



Mandelbrotのテストイング

❖ テスト項目5種類

- ❖ 履歴情報の大きさが仕様に合致すること
- ❖ 履歴情報における表示画面位置(履歴情報の第二成分)が仕様の'[Xx]'の出現位置に合致すること
- ❖ 履歴の底は初期画面であること
- ❖ 履歴情報に含まれる表示範囲について包含関係が成立すること.
- ❖ 履歴情報のうちドラッグされたばかりの領域(X)のみが変更されていること

Mandelbrotテストのまとめ

- ❖ ソフトウェアのテスト作業は想像力と創造性をともに要求される高度に知的なパズル
- ❖ Mandelbrotでお腹いっぱいになったので、近所の八百屋で見つけたロマネスコをいただきました。
- ❖ デザートが欲しい人はここが面白いですよ。

少し早いお年玉

- ❖ package lx10

- ❖ dragdemo.scala

- ❖ rectangles.scala

- ❖ shapes.scala

- ❖ 次回のテーマ（考えておいて下さい）

- ❖ shapes.scalaの実装をモデルとビューに分離すること

- ❖ shapes.scalaのテストケースを用意すること

