8.2 複素数

これまでに、狭いながらも庭の隅をいじり回してきたね。大した道具を手にしていたわけじゃないので、ちょっとしたことでも苦労の連続だった。もう少しマシな道具があれば庭いじりも楽になるだろうに。そう、**Haskell** には便利な道具がある。モジュールだ。以前使ったことがあるよね。

たとえば分数の計算、そうだな $\dfrac{1}{3}+\dfrac{2}{5}$ でもやってみよう。それなら

(ghci env.)

```
Prelude> 1/3 + 2/5
0.7333333333333334
```

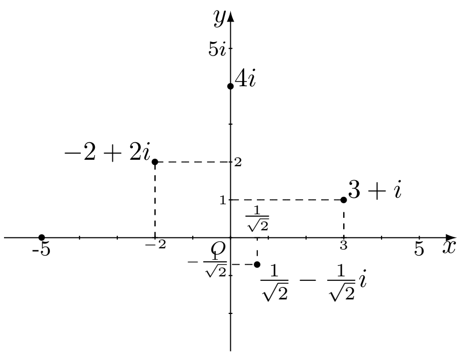
で十分だろう。いや、そうじゃない。分数のまま答えを出したいんだ。そんなときは `Data.Ratio` モジュールをインポートするとよい。それで分数が分数のまま扱える。

(ghci env.)

```
Prelude> import Data.Ratio
Prelude Data.Ratio> 1%3 + 2%5
11 % 15
```

ちょっと見づらいかも。**Haskell** では分数 $\dfrac{a}{b}$ の表記は `a%b` である。したがって `11 % 15` は $\dfrac{11}{15}$ を意味する。計算は合っているよね。で、複素数の番だ。

ところで実数と虚数は数直線上に表せるが、複素数とは実際どんな数なのだろう。複素数は数直線ではなく、複素平面上に表す。



たとえば、この中のいくつかを使って **Haskell** に計算してもらおう。そのためには `Data.Complex` をインポートしなくてはならない。もし `Data.Ratio` をインポートしているときに `Data.Complex` もインポートすれば

(ghci env.)

```
Prelude Data.Ratio> import Data.Complex
Prelude Data.Ratio Data.Complex>
```

となるはずだ。ひゃほー。分数も複素数も使えるの？　でも、その前に複素数ってどう表すんだろう。**Haskell** で複素数 $2+3i$ は `2:+3` と表す。つまり演算子 “`:+`” が複素数を意味している。i は虚数単位として認識してくれない。つまり `2:+3` は 2 と 3 の演算でなく、複素数 $2+3i$ を表す。

(ghci env.)

```
Prelude Data.Ratio Data.Complex> 2:+3
2 :+ 3
Prelude Data.Ratio Data.Complex> (2:+3)-(1:+5)
1.0 :+ (-2.0)
```

単に $2+3i$ を入力すれば `2 :+ 3` となるが、$(2+3i)-(1+5i)$ の計算を試すと `1.0 :+ (-2.0)` だ。計算は単精度か倍精度の実数で扱われるのかな。だったら $\dfrac{1}{3}+\dfrac{2}{5}i$ はどうなるんだろう。

(ghci env.)

```
Prelude Data.Ratio Data.Complex> 1%3 :+ 2%5
1 % 3 :+ 2 % 5
Prelude Data.Ratio Data.Complex> (1%3 :+ 2%5)-(1%2 :+ 1%5)

<interactive>:16:1: error:
    • Non type-variable argument in the constraint: RealFloat (Ratio a)
      (Use FlexibleContexts to permit this)
    • When checking the inferred type
        it :: forall a.
              (RealFloat (Ratio a), Integral a) =>
              Data.Complex.Complex (Ratio a)
```

やっぱりね。分数と複素数が使えるといっても、同時に使えるわけじゃないんだ。それでも複素数が扱えることは分かった。問題は、複素数はどんな規則で計算されるかということだ。規則といっても、一般的な複素数 $a+bi$ と $c+di$ で考えれば

$$
\begin{aligned}
(a+bi)+(c+di) &= (a+c)+(b+d)i \\
(a+bi)-(c+di) &= (a-c)+(b-d)i \\
(a+bi)(c+di) &= (ac-bd)+(ad+bc)i
\end{aligned}
$$

であることはすぐ分かるし、割り算の場合は、

$$
\begin{aligned}
(a+bi)/(c+di) &= \frac{(a+bi)(c-di)}{(c+di)(c-di)} \\
&= \frac{ac+bd}{c^2+d^2}+\frac{-ad+bc}{c^2+d^2}i
\end{aligned}
$$

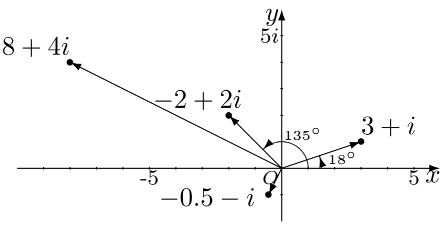
のように、分子・分母に $c-di$ を掛ければ結果はやはり複素数だ。ちなみに、$c+di$ と $c-di$ のように虚部が符号違いの複素数を互いに共役（きょうやく）という。共役複素数のペアは複素数の

計算で重要な位置を占めているが、ここではとくに取り上げない。念のため、計算規則が **Haskell** で成り立っていること確認しておこう。

(ghci env.)

```
Prelude Data.Ratio Data.Complex> (3:+1) + ((-2):+2)
1.0 :+ 3.0
Prelude Data.Ratio Data.Complex> (3:+1) * ((-2):+2)
(-8.0) :+ 4.0
Prelude Data.Ratio Data.Complex> (3:+1) / ((-2):+2)
(-0.5) :+ (-1.0)
```

気をつけないといけないことがひとつある。**Haskell** では、負の値は必ず (-2) のように括弧を付ける必要があることだ。そうしないと **Haskell** に怒られてしまうから。でも、問題は数値がどうなるかではなく、数値が複素平面のどこに現れるか、なのである。たとえば $(3+i)(-2+2i)=-8+4i$ だ。どうして変な場所に現れてしまうのだろう。



結論を言おう。複素数は "点" ではなく "**ベクトル**" である。ベクトルは大きさと向きがある矢線なので、足したり引いたりすることは、矢線を継ぎ足すことである（引き算は逆向きにして継ぎ足す）。それは、実際に矢線を描いてみれば分かる。

ベクトルの大きさとは矢線の長さである。三平方の定理によって $a+bi$ の大きさは $\sqrt{a^2+b^2}$ で求められる。具体的に $3+i$ の大きさは $\sqrt{3^2+1^2}=\sqrt{10}$、$-2+2i$ の大きさは $\sqrt{(-2)^2+2^2}=2\sqrt{2}$ である。だから、$(3+i)(-2+2i)=-8+4i$ となるのは大きさの観点から見て正しい。なぜなら、$-8+4i$ の大きさは $\sqrt{(-8)^2+4^2}=4\sqrt{5}$ なので、たしかに $\sqrt{10}\times 2\sqrt{2}=4\sqrt{5}$ だからだ。割り算も同様にして、$-0.5-i$ の大きさは $\dfrac{\sqrt{5}}{2}$ なので、たしかに $\sqrt{10}/(2\sqrt{2})=\dfrac{\sqrt{5}}{2}$ だ。

では、複素数が現れる "場所" はどうして決まるんだろう。これにはベクトルが持つ**偏角**が関係している。偏角とは、ベクトルが x 軸の正の方向となす角をいう。これは少し分かりづらいし、正確に求めるには三角関数を用いて計算しなくてはならない。ここでは大まかに見ておこう。矢線を引けば大体の雰囲気がつかめると思うが、$3+i$ の偏角はおよそ $18°$ で、$-2+2i$ の偏角はちょうど $135°$ である。そこで、2 つの複素数の積はそれぞれの偏角の和の位置に、商はそれぞれの偏角の差の位置に現れる。$3+i$ と $-2+2i$ の偏角の和はおよそ $153°$、偏角の差はおよそ $-117°$ である。これらは、積 $-8+4i$ の偏角と商 $-0.5-i$ の偏角に等しいのである。

話は変わるが、実数の単位は何であろうか。1 である。では、虚数の単位は？　もちろん i である。ならば、複素数の単位は何であろう。$1+i$ では単位とは言えない。というのは、単位とは大きさが 1 のものを指すからだ。ベクトル $1+i$ と同等で大きさが 1 となると、それは $\dfrac{1}{\sqrt{2}}+\dfrac{1}{\sqrt{2}}i$ である。さて、再び強引な話になるけれど、単位 1 の符号は $+$ で単位 i の符号は i であるとは前に言ったね。じゃあ、単位 $\dfrac{1}{\sqrt{2}}+\dfrac{1}{\sqrt{2}}i$ の符号って何だろう？　実際は、符号という考えはおかしいのだが、i の符号をそのまま i と見たので、$\dfrac{1}{\sqrt{2}}+\dfrac{1}{\sqrt{2}}i$ はそのまま符号扱いしておこう。すると符号の積が

$$\left(\frac{1}{\sqrt{2}}+\frac{1}{\sqrt{2}}i\right)\times\left(\frac{1}{\sqrt{2}}+\frac{1}{\sqrt{2}}i\right)=i$$

となる。このことから、-1 の平方根が i であるように、i の平方根は $\dfrac{1}{\sqrt{2}}+\dfrac{1}{\sqrt{2}}i$ であると言ってよいだろう。つまり、1 の平方根（-1）の平方根（i）の平方根が $\dfrac{1}{\sqrt{2}}+\dfrac{1}{\sqrt{2}}i$ なのである。実際、

(ghci env.)

```
Prelude Data.Ratio Data.Complex> let c = 1/sqrt 2:+1/sqrt 2
Prelude Data.Ratio Data.Complex> c^8
0.9999999999999991 :+ (-0.0)
```

のように、$c=\dfrac{1}{\sqrt{2}}+\dfrac{1}{\sqrt{2}}i$ とおいて c^8 を計算すると、精度の都合でぴったりとはいかないが、$c^8=1$ を示唆している。

まとめるとこうだ。$(-)^2=(+)$ とは計算上の約束であり、$(?)^2=(-)$ となる符号はないものとしてきた。逆の見方をすれば、$(+)$ は $(-)$ どうしの積にできるが、$(-)$ はそれ以上分解できないものと考えてきたわけである。しかし定義を拡張することで、$(-)=(i)^2$ との約束を取り付けることができた。これは、$(-)$ は (i) の積に分解できると見なしてよい。まるで、これ以上分解できないと思っていた原子が、陽子と中性子に分けられることが分かったように。

さらに、(i) を符号と見ると、$\left(\dfrac{1}{\sqrt{2}}+\dfrac{1}{\sqrt{2}}i\right)^2=(i)$ だから、(i) でさえ別の符号どうしの積にできるともいえる。ただし、複素数で表される数を符号とみなすには無理があるけれど。このことは、原子の根源であると思った陽子でさえ、さらにいくつかのクォークに分けられることに似ている。あろうことか、複素数はさらにその上を行く。$\left(\dfrac{\sqrt{2+\sqrt{2}}}{2}+\dfrac{\sqrt{2-\sqrt{2}}}{2}i\right)^2=\dfrac{1}{\sqrt{2}}+\dfrac{1}{\sqrt{2}}i$ になるからだからだ。そして、さらにさらに $\dfrac{\sqrt{2+\sqrt{2}}}{2}+\dfrac{\sqrt{2-\sqrt{2}}}{2}i$ でさえ... ということが際限なく続くのである。でも、この行き着く先は意外な場所である。それは $1+0i$ である。

$+1=(-1)^2$、$-1=i^2$、$i=\left(\dfrac{1}{\sqrt{2}}+\dfrac{1}{\sqrt{2}}i\right)^2$、$\dfrac{1}{\sqrt{2}}+\dfrac{1}{\sqrt{2}}i=(\cdots)^2$、... というように、深く深く突き詰めた先が、もとの $+1$ 戻ってしまうなんて。複素数には不思議な関連があるものだ。