cpx.scm

Dmytro Bondarenko

Seminar in Konstruktive Analysis
Wintersemester 2023/2024

1 Einbettung von $\mathbb{R}$ in $\mathbb{C}$, Definition von $\mathbb{C}$

Wir definieren die komplexen Zahlen durch den Konstruktor:

$$\texttt{CpxConst:}\ \mathbb{R} \Rightarrow \mathbb{R} \Rightarrow \mathbb{C}$$

- Totalität: $\forall x, y(\ \texttt{TotalRea}\ x \to \texttt{TotalRea}\ y \to \texttt{TotalCpx}\ (x, y)$

Zur Einbettung:

$$\texttt{RealToCpx} : \mathbb{R} \to \mathbb{C},\ x \mapsto (x; 0).$$

Wollen, analog zu rationalen, reellen Zahlen, auf die Bestandteile des Konstruktors zugreifen:

- `RealPart`: `Re` $(x; y) = x$
- `ImagPart`: `Im` $(x; y) = y$

2 Arithmetik

Ähnlich bei reellen Zahlen definieren wir Operationen wie +, - mithilfe von `computation rules`:

- `CpxPlus`: $(x_1; y_1) + (x_2; y_2) := (x_1 + x_2; y_1 + y_2)$
- `CpxTimes`: $(x_1; y_1) * (x_2; y_2) := (x_1 * x_2 - y_1 * y_2; x_1 * y_2 + x_2 * y_1)$
- `CpxUMinus`: $-(x; y) := (-x; -y)$
- `CpxMinus`: $(x_1; y_1) - (x_2; y_2) := (x_1; y_1) + (-(x_2; y_2))$
- `CpxUDiv`: $\frac{1}{(x;y)} := (\frac{x}{x^2+y^2}; \frac{-y}{x^2+y^2})$

Insbesondere:

- $(0; 1) * (0; 1) = (0 - 1; 0 + 0) = (-1; 0) = -1 \in \mathbb{R}$
- $\frac{1}{(0;1)} = (0; \frac{-1}{1}) = -(0; 1)$

3 Induktive Prädikaten

$$\forall z(\texttt{Re}\ z \in \mathbb{R} \to \texttt{Im}\ z \in \mathbb{R} \to z \in \mathbb{C})$$

In Minlog nutzt man oft `cases`, das heißt eine Zahl wird in die Konstruktorteile zerlegt, und die Aussage wird dadurch eine Eben "einfacher". In diesem Fall wird es hilfreich, wenn man, um einen Satz zu beweisen, der $z \in \mathbb{C}$ enthält, diese Zahl als (`Re` z; `Im` z) darstellt. Folgende Sätze sind deswegen nützlich:

- `CpxConstrToReal1`: $\forall x, y((x; y) \in \mathbb{C} \Rightarrow x \in \mathbb{R})$
- `CpxConstrToReal2`: $\forall x, y((x; y) \in \mathbb{C} \Rightarrow y \in \mathbb{R})$

4 Gleichheit

Wir definieren Gleichheit von komplexen Zahlen durch die Gleichheit der reellen Zahlen. Dazu: ein induktives Prädikat `CpxEq`, was später in Minlog mit einem Zeichen `-=-` ersetzt wird.
Einführungsregel `CpxEqIntro`:

$$\forall z_1, z_2(\texttt{Re}\ z_1 = \texttt{Re}\ z_2 \to \texttt{Im}\ z_1 = \texttt{Im}\ z_2 \to z_1 = z_2)$$

Wie bei jeder Einführungsregel, gibt es auch hier Eliminationsaxiome:

- `CpxEqElim0`: $\forall z_1, z_2(z_1 = z_2 \to \texttt{Re}\ z_1 = \texttt{Re}\ z_2)$
- `CpxEqElim1`: $\forall z_1, z_2(z_1 = z_2 \to \texttt{Im}\ z_1 = \texttt{Im}\ z_2)$

Ex-Falso-Quodlibet für die Gleichheit:

$$\texttt{EfCpxEq:}\ F \to \forall z_1, z_2(z_1 = z_2)$$

Da die Gleichheit für die komplexen Zahlen eine Äquivalenzrelation ist, lassen sich hier auch die 3 Eigenschaften einfach beweisen. Man zerlegt komplexe Zahlen in die reellen Anteile und nutzt aus, daß die reelle Gleichheit ebenso eine Äquivalenzrelation ist. Der Beweis ist trivial. In Minlog wird es wie folgt formuliert:

- Reflexivität `CpxEqRefl`: $\forall z(z \in \mathbb{C} \to z = z)$
- Symmetrie `CpxEqSym`: $\forall z_1, z_2(z_1 \in \mathbb{C} \to z_2 \in \mathbb{C} \to z_1 = z_2 \to z_2 = z_1)$
- Transitivität `CpxEqTrans`:
  $\forall z_1, z_2, z_3(z_1, z_2, z_3 \in \mathbb{C} \to z_1 = z_2 \to z_2 = z_3 \to z_1 = z_3)$

Weitere nützliche Eigenschaft der komplexen Gleichtheit:
Verträglichkeit mit `CpxEqPlus`, `CpxEqTimes` und `RealEq` (ist natürlich auch sinvoll für `CpxEqMinus` etc., wurde hier erstmal nicht behandelt):

- `RealEqToCpxEq`:
  $\forall x, y(x \overset{\mathrm{R}}{=} y \to x \overset{\mathrm{C}}{=} y)$
- `CpxPlusCompat`:
  $\forall z_1, z_2, z_3, z_4(z_1 \in \mathbb{C} \to z_2 \in \mathbb{C} \to z_3 \in \mathbb{C} \to z_4 \in \mathbb{C} \to$
  $\to z_1 = z_2 \to z_3 = z_4 \to z_1 + z_3 = z_2 + z_4)$
- `CpxTimesCompat`:
  $\forall z_1, z_2, z_3, z_4(z_1 \in \mathbb{C} \to z_2 \in \mathbb{C} \to z_3 \in \mathbb{C} \to z_4 \in \mathbb{C} \to$
  $\to z_1 = z_2 \to z_3 = z_4 \to z_1 * z_3 = z_2 * z_4)$

5 Abgeschloßenheit

Eine offensichtliche, aber dennoch wichtige Eigenschaft, die in Minlog explizit bewiesen werden muß: Summe, Multiplikation, etc. von komplexen Zahlen liefert wieder eine komplexe Zahl. Das heißt, die komplexen Zahlen sind abgeschloßen unter den früher definierten arithmetischen Operationen.

- `CpxPlusCpx`:
  $\forall x, y(x \in \mathbb{C} \to y \in \mathbb{C} \to x + y \in \mathbb{C})$
- `CpxUMinusCpx`, `CpxMinusCpx`, `CpxTimesCpx`, `CpxUDivCpx`: ähnlich.

6 Gleichheit/Operationen Eigenschaften

Nachdem man genug Operationen und die Gleichheit dafür bereits entwickelt hat, macht es Sinn, die einfachen Schlußfolgerungen zu beweisen, indem man jeweils die gleiche Eigenschaft für die reellen Zahlen ausnutzt.

- Kommutativität von Plus, `CpxPlusComm`:
  $\forall z_1, z_2(z_1 \in \mathbb{C} \to z_2 \in \mathbb{C} \to z_1 + z_2 = z_2 + z_1)$
- Assoziativität von Plus, `CpxPlusAssoc`:
  $\forall z_1, z_2, z_3(z_1 \in \mathbb{C} \to z_2 \in \mathbb{C} \to z_3 \in \mathbb{C} \to z_1 + z_2 + z_3 = z_1 + (z_2 + z_3))$
- `CpxPlusZero`: $\forall z(z \in \mathbb{C} \to z + 0 = z)$
- `CpxTimesOne`: $\forall z(z \in \mathbb{C} \to z * 1 = z)$

Kommutativität und Assoziativität von `CpxTimes` lassen sich identisch formulieren und beweisen.

7 Konjugation

Der erster wichtiger Punkt, der sich bei komplexen Zahlen von reellen bzw. rationalen Zahlen unterscheidet: Konjugation. Dazu definieren wir in Minlog ein `program-constant "CpxConj"`, das eine komplexe Zahl nimmt und eine komplexe Zahl als Ergebnis liefert. `computation-rules` sieht wie folgt aus:

```
CpxConj (x;y) = (x;-y)
```

Hier, wir verwenden die gewöhnliche Schreibweise für die komplexe Konjugation, nämlich $\overline{z}$.
Wie bei jedem neuen `program-constant`, beweist man auch hier die Totalität von `CpxConj`. Einige wichtige Eigenschaften:

- Reelle Anteile bleiben bei Konjugation unverändert, `CpxConjRealPart`:
  $\forall z(z \in \mathbb{C} \to \texttt{Re}\ \overline{z} = \texttt{Re}\ z)$
- Doppelte Konjugation, `CpxConjConj`: $\forall z(z \in \mathbb{C} \to \overline{\overline{z}} = z)$

- Summe von zwei konjugierten Zahlen, `CpxConjSum`:
  $\forall z_1, z_2(z_1 \in \mathbb{C} \to z_2 \in \mathbb{C} \to \overline{z_1 + z_2} = \overline{z_1} + \overline{z_2})$
- Produkt von zwei konjugierten Zahlen, `CpxConjTimes`:
  $\forall z_1, z_2(z_1 \in \mathbb{C} \to z_2 \in \mathbb{C} \to \overline{z_1 * z_2} = \overline{z_1} * \overline{z_2})$

Letzendlich definieren wir in diesem Kapitel die imaginäre Einheit i.

```
(add-program-constant "ImagUnit" (py "cpx"))
```

Die Berechnungsregel: `ImagUnit` $= (0;1)$. Hier bezeichnen wir die imaginäre Einheit, wie üblich, mit i.
Von früher haben wir bereits: $i^2 = -1$; $\frac{1}{i} = -i$.
Der nächste natürliche Schritt: eine komplexe Zahl `(x;y)` als `x+iy` zu schreiben.
`CpxImagForm`: $\forall x, y((x;y) = x + i * y)$
Beweis:

$$\begin{aligned} x + i * y &= (x;0) + (0;1) * (y;0) \\ &= (x;0) + (0 * y - 1 * 0; 0 * 0 + 1 * y) \\ &= (x;0) + (0;y) \\ &= (x;y) \end{aligned} \tag{1}$$

Zwei weitere wichtige Eigenschaften, die sich auf eine gewöhnliche Weise einfach beweisen lassen:

- `CpxPlusConj`: $\forall z(\texttt{Re}\ z = \frac{1}{2} * (z + \overline{z}))$
- `CpxMinusConj`: $\forall z(\texttt{Im}\ z = \frac{1}{2} * (z - \overline{z}))$

Die letzte Eingenschaft in diesem Kapitel, die wir später für die Trigonometrie brauchen werden:

$$\texttt{CpxTimesConjCpx}: \forall z(z * \overline{z} = \texttt{Re}^2 z + \texttt{Im}^2 z)$$

8 Wurzel

Bevor wir den Absolutbetrag einführen, muß der quadratische Wurzel behandelt werden, der sich durch Herron - Verfahren konstruktiv definieren lässt. Hier werden wir nur die Grundeigenschaften auflisten, um die später nutzen zu können.

- $\forall x(0 \leq \sqrt{x})$
- $\forall x(\sqrt{x} = 0 \Leftrightarrow x = 0)$
- $\forall x, y(x = y \Rightarrow \sqrt{x} = \sqrt{y})$
- $\forall x, y(\sqrt{x} = \sqrt{y} \Rightarrow x = y)$
- $\forall x, y(\sqrt{xy} = \sqrt{x}\sqrt{y})$
- $\forall x(\sqrt{x}\sqrt{x} = x)$

9 Absolutbetrag

Definiere $|z| := \sqrt{z * \overline{z}}$. Da $\sqrt{z * \overline{z}}$ von Minlog nicht als reelle Zahl angesehen wird, haben wir früher bewiesen: $\forall z(z * \overline{z} = \mathtt{Re}^2 z + \mathtt{Im}^2 z)$.

Jetzt: definiere Absolutbetrag

$$|\cdot| : \mathbb{C} \to \mathbb{R},\ z \mapsto \sqrt{\mathtt{Re}^2 z + \mathtt{Im}^2 z}$$

Eigenschaften: $\forall z, z_1, z_2 \in \mathbb{C}$

- $|z| = |\overline{z}|$
- $0 \leq |z|$: folgt aus der Nicht-Negativität des quadratischen Wurzels
- $|z| = 0 \Leftrightarrow z = 0$
- $|z_1 * z_2| = |z_1| * |z_2|$
- $|z_1 + z_2| \leq |z_1| + |z_2|$

10 Ausblick: Exponentialreihe und Trigonometrie

10.1 Exponentialfunktion

Definieren rekursiv natürliche Potenz einer komplexen Zahl in Minlog:

$$\mathtt{CpxPower}\colon \mathbb{C} \to \mathbb{N} \to \mathbb{C},\ z^0 = 1,\ z^{n+1} = z * z^n$$

Weiter definieren wir die komplexe Exponentialreihe analog zu der reellen Exponentialreihe.

1. Folge $(z_n)_{n\in\mathbb{N}} \Rightarrow$ zerlege in $(\mathtt{Re}\ z_n)_{n\in\mathbb{N}}$ and $(\mathtt{Im}\ z_n)_{n\in\mathbb{N}}$
2. $(z_n)_{n\in\mathbb{N}} \subset \mathbb{C} \Rightarrow (\mathtt{Re}\ z_n)_{n\in\mathbb{N}}, (\mathtt{Im}\ z_n)_{n\in\mathbb{N}} \subset \mathbb{R}$
3. $\mathtt{CpxLim}\ (z_n)_{n\in\mathbb{N}}\ M = (\mathtt{RLim}\ (\mathtt{Re}\ z_n)_n\ M; \mathtt{RLim}\ (\mathtt{Im}\ z_n)_n\ M)$, das heißt, Limes einer komplexen Folge ist eine komplexe Zahl, die als Bestandteile jeweils die Limiten von $(\mathtt{Re}\ z_n)_n$ und $(\mathtt{Im}\ z_n)_n$ hat.

Zu dem Modul der Konvergenz: die Aussage oben muß stimmen, da, falls die komplexe Folge mit einem Modul M konvergiert, muß es reichen, um auch die Konvergenz von Reell- bzw. Imaginärfolgen mit demselben Modul zu erhalten. Andersrum: konvergiert die Folge der reellen Anteilen mit einem Modul M und die Folge der imaginären Anteilen mit einem Modul N, so muß die komplexe Folge mit Modul $max(M, N)$ gegen die Zahl konvergieren, die die M- und N-Limiten als Bestandteile hat.
Nächste Definition, komplett analog zu reellen Zahlen:

$$\mathtt{CpxSum} + \mathtt{CpxSumCpx}(\text{Summe von komplexen Zahlen ist komplex})$$

Wichtige Verallgemeinerung: Folge von endlichen komplexen Summen:

$$\texttt{CpxESeq}\colon \mathbb{C} \to \mathbb{N} \to \mathbb{C}$$

Endlich: definieren die Exponentialfunktion $\mathbb{C} \to \mathbb{C}$, `CpxE`, als Limes der Konvergenz einer Folge von endlichen komplexen Summen `CpxESeq`.
Eigenschaften der komplexen Exponentialfunktion:

1. `CECpx`: $\forall z(z \in \mathbb{C} \to e^z \in \mathbb{C})$
2. `CpxExpConj`: $\forall z(z \in \mathbb{C} \to e^{\overline{z}} = \overline{e^z})$
3. Funktionalgleichung `CpxExpFuncEq`:
   $\forall z_1, z_2(z_1, z_2 \in \mathbb{C} \to e^{z_1+z_2} = e^{z_1} * e^{z_2})$
4. Verträglichkeit `CpxExpCompat`
   (Bemerkung: nicht injektiv, da es Perioden gibt):
   $\forall z_1, z_2(z_1, z_2 \in \mathbb{C} \to z_1 = z_2 \to e^{z_1} = e^{z_2})$
5. `CpxExpZero`: $e^0 = 1$
6. `CpxExpImagxAbsValue1`: $\forall x \in \mathbb{R}$: $|e^{ix}| = 1$
   Beweis: $|e^{ix}|^2 \overset{\text{Def.}}{=} e^{ix} * \overline{e^{ix}} \overset{2}{=} e^{ix} * e^{\overline{ix}} = e^{ix} * e^{-ix} \overset{3}{=} e^0 \overset{5}{=} 1.$

10.2 Trigonometrie

Wir definieren Kosinus und Sinus als Reell- bzw. Imaginäranteil einer Exponentialreihe:

- $cos: \ \mathbb{R} \to \mathbb{R}, \ x \mapsto \texttt{Re}\ e^{ix}$
- $sin: \ \mathbb{R} \to \mathbb{R}, \ x \mapsto \texttt{Im}\ e^{ix}$

Als erstes müßen die Totalitäten von `cos` und `sin` bewiesen werden, und das folgt leicht aus den Totalitäten von `Re`, `Im`, `CpxExp` etc. Weitere Eigenschaften:

- $\forall x \in \mathbb{R}: \ e^{ix} = (cos\ x; sin\ x)$ (folgt sofort aus der Definition)
- $\forall x \in \mathbb{R}: \ e^{ix} = cos\ x + i * sin\ x$
- $\forall x \in \mathbb{R}: \ cos^2\ x + sin^2\ x = 1$ (folgt aus: $|e^{ix}| = 1$)

Explizite Darstellung von sin und cos:

- $cos\ x = \frac{1}{2} * (e^{ix} + e^{-ix})$ (folgt aus: $\texttt{Re}\ z = \frac{1}{2} * (z + \overline{z})\ \forall z \in \mathbb{C}$)
- $sin\ x = \frac{1}{2i} * (e^{ix} - e^{-ix})$ (folgt aus: $\texttt{Im}\ z = \frac{1}{2i} * (z - \overline{z})\ \forall z \in \mathbb{C}$)