

Vollständigkeit der reellen Zahlen

Seminar Konstruktive Analysis, Mathematisches Institut, LMU

Wintersemester 2024/25

Definition

Eine Folge $(x_n)_{n \in \mathbb{N}}$ reeller Zahlen ist eine **Cauchyfolge** mit Modul $M: \mathbb{Z}^+ \rightarrow \mathbb{N}$ wenn

$$|x_n - x_m| \leq \frac{1}{2^p} \quad \text{für } n, m \geq M(p),$$

und **konvergiert** mit Modul $M: \mathbb{Z}^+ \rightarrow \mathbb{N}$ gegen eine reelle Zahl y , wenn

$$|x_n - y| \leq \frac{1}{2^p} \quad \text{für } n \geq M(p).$$

Der Grenzwert y ist eindeutig bestimmt.

Lemma (L1, RatCauchyConvMod)

Jede modulierte Cauchyfolge rationaler Zahlen konvergiert mit demselben Modul gegen die von ihr dargestellte reelle Zahl.

- Sei $x := ((a_n)_n, M)$ eine reelle Zahl. Zu zeigen ist

$$|a_n - x| \leq \frac{1}{2^p} \quad \text{für } n \geq M(p).$$

- Fixiere $n \geq M(p)$. Es genügt zu zeigen

$$|a_n - a_m| \leq \frac{1}{2^p} \quad \text{für } m \geq M(p).$$

- Das gilt nach Annahme.

Lemma (L2, RealLimReal)

Für jede Cauchyfolge reeller Zahlen findet man eine reelle Zahl gegen die sie konvergiert.

1. Sei $(x_n)_{n \in \mathbb{N}}$ eine Cauchyfolge reeller Zahlen mit Modul M ; etwa $x_n := ((a_{nl})_l, N_n)$.
2. Für jedes $n \in \mathbb{N}$ und jedes p gilt nach RatCauchyConvMod

$$|x_n - a_{nl}| \leq \frac{1}{2^p} \quad \text{für alle } l \geq N_n(p).$$

3. Sei

$$b_n := a_{nN_n(n)}$$

Dann

$$|x_n - b_n| \leq \frac{1}{2^n} \quad \text{for all } n \in \mathbb{N}$$

nach dem Fall $l = N_n(n)$ von 2.

Wir hatten

$$|x_n - b_n| \leq \frac{1}{2^n} \quad \text{for all } n \in \mathbb{N}.$$

Also

$$|b_n - b_m| \leq |b_n - x_n| + |x_n - x_m| + |x_m - b_m| \leq \frac{1}{2^n} + \frac{1}{2^{q+1}} + \frac{1}{2^m} \leq \frac{1}{2^q}$$

für alle $n, m \geq \max(M(q+1), q+2)$.

Das heißt: $y := (b_n)_n$ ist eine Cauchyfolge mit Modul

$$K(q) := \max(M(q+1), q+2).$$

Lemma (L3, RealCauchyConvMod)

Jede modulierte Cauchyfolge reeller Zahlen konvergiert mit demselben Modul gegen ihren Limes.

- Sei $(x_n)_n$ eine Cauchyfolge reeller Zahlen mit Modul M , d.h.

$$|x_n - x_m| \leq \frac{1}{2^p} \quad \text{für } n, m \geq M(p).$$

- Sei y ihr Limes, also

$$|x_n - y| \leq \frac{1}{2^q} \quad \text{für } n \geq K(q).$$

- Wir zeigen

$$|x_n - y| \leq \frac{1}{2^p} \quad \text{für } n \geq M(p).$$

- Fixiere $n \geq M(p)$. Sei $q \in \mathbb{Z}^+$. Dann

$$\begin{aligned} |x_n - y| &\leq |x_n - x_m| + |x_m - y| \quad \text{für } m \geq M(p), K(q) \\ &\leq \frac{1}{2^p} + \frac{1}{2^q}. \end{aligned}$$

Die Behauptung folgt, da dies für alle q gilt.

```
;; RealCompleteAux1
(set-goal "all ass,Ns,xs,bs(
  all n xs n eqd RealConstr(ass n)(Ns n) ->
  all n Real(xs n) ->
  all n bs n=ass n(Ns n(cNatPos n)) ->
  all n abs(bs n+ ~(xs n))<=<=(1#2**n))")
```

```
;; RealCompleteAux2
(set-goal "all ass,Ns,xs,M,bs,K(
  all n Real(xs n) ->
  all p,n,m(M p<=n -> M p<=m -> abs(xs n+ ~(xs m))<=(1#2**p
  all n xs n eqd RealConstr(ass n)(Ns n) ->
  all n bs n=ass n(Ns n(cNatPos n)) ->
  all p K p=M(PosS p)max PosS(PosS p) ->
  all p,n,m(K p<=n -> K p<=m -> abs(bs n+ ~(bs m))<=(1#2**p
```

```
(add-program-constant "RealLim"  
  (py "(nat=>rea)=>(pos=>nat)=>rea"))  
  
(add-computation-rules  
  "RealLim xs M"  
  "RealConstr([n](xs n)seq((xs n)mod(cNatPos n)))  
  ([p]M(PosS p)max PosS(PosS p))")
```

```
;; L3 (RealCompleteAux3 oder RealCauchyConvMod)
(set-goal "all ass,Ns,xs,M,bs,K,x(
  all n xs n eqd RealConstr(ass n)(Ns n) ->
  RealCauchy xs M ->
  all n bs n=ass n(Ns n(cNatPos n)) ->
  all q K q=M(PosS q)max PosS(PosS q) ->
  x==RealConstr bs K ->
  all p,n(M p<=n -> abs(xs n+ ~x)<=(1#2**p))))")
```

```
;; L4 (RealComplete)
(set-goal "all xs,M(RealCauchy xs M ->
  all p,n(M p<=n -> abs(xs n+ ~(cRLim xs M))<=(1#2**p)))")
```

```
;; L5 (RealCompleteCor)
(set-goal "all xs,M(RealCauchy xs M ->
  RealConvLim xs(cRLim xs M)M)")
```

L4 (RealComplete) und L5 (RealCompleteCor) sagen dasselbe aus wie L3, nur mit Verwendung von cRLim.

RealLim und cRLim

```
(add-program-constant "RealLim"  
  (py "(nat=>rea)=>(pos=>nat)=>rea"))
```

```
(add-computation-rules  
  "RealLim xs M"  
  "RealConstr([n](xs n)seq((xs n)mod(cNatPos n)))  
    ([p]M(PosS p)max PosS(PosS p))")
```

Problem: (RealLim xs M) faltet beim Normalisieren immer aus.

Ausweg:

Beweise einen Existenzsatz RLim und “animiere” RLim . Damit wird cRLim definiert, als der rechnerische Gehalt des Existenzsatzes.

```
;; RLim
(set-goal "all xs,M ex1 x x eqd RealLim xs M")
(assume "xs" "M")
(intro 0 (pt "RealLim xs M"))
(intro 0)
;; Proof finished.
;; (cp)
(save "RLim")
```

```
(animate "RLim")

;; RLimExFree
(set-goal "all xs,M cRLim xs M eqd RealLim xs M")
(assume "xs" "M")
(ng #t)
(intro 0)
;; Proof finished.
;; (cp)
(save "RLimExFree")

(deanimate "RLim")
```

Jetzt kann man das deanimierte `cRLim` anstelle von `RealLim` verwenden. Es faltet aus genau dann wenn `RLim` animiert ist.