# Konstruktive Mathematik, die Doppelnegationsübersetzung und Continuations

Ingo Blechschmidt <iblech@speicherleck.de>, Universität Augsburg

*Wie können wir das Minimum einer unendlichen Liste von natürlichen Zahlen bestimmen?* Wenn wir die Liste nicht überschauen können, sind wir auf folgenden Trick angewiesen. Wir bluffen und proklamieren einfach das erste Element der Liste als Minimum. Zufälligerweise könnte das sogar wirklich das kleinste Listenelement sein. Wenn uns aber jemand herausfordert und uns ein kleineres Element präsentiert, so springen wir in der Zeit zurück und ändern die Geschichte so, als ob wir von vornherein *dieses* Element als Minimum proklamiert hätten. Sollte auch dieser Bluff auffliegen, wiederholen wir das Prozedere.

Ist das geschummelt? Ja, in einem gewissen Sinn schon. Es steckt allerdings mehr dahinter: Zum einen funktioniert dieses Verfahren „von innerhalb der Continuation-Monade betrachtet“ taddellos und kann in gewissen Situationen durchaus nützlich sein. Obacht ist nur geboten, wenn man „die Continuation-Monade verlässt“. Zum anderen entstammt das Verfahren nicht dem übertriebenen Ehrgeiz eines Poker-Fans, sondern ergibt sich *maschinell* durch „Extraktion algorithmischen Inhalts“ aus einem mathematischen Beweis der einfachen Tatsache, dass jede unendliche Liste von natürlichen Zahlen ein Minimum enthalten muss.

Im Vortrag werden wir alle Floskeln dieses Absatzes verstehen. Haskell-Code wird die Sachverhalte illustrieren. Wir werden auch einen praktischen Anwendungsfall aus der Kryptographie und Zahlentheorie diskutieren (je nach Wunsch während des Vortrags oder durch ausführliche schriftliche Notizen), der „Bestimmung von modularen Inversen in Restklassenringen“, in der die vorgestellten Techniken eine erhebliche Effizienzsteigerung ermöglichen.

## Notwendiger Exkurs: Was ist konstruktive Mathematik?

Den Unterschied zur gewöhnlichen, klassischen Mathematik begreift man am besten an einem Beispiel. Eine Zahl heißt genau dann *rational*, wenn sie sich als Bruch zweier ganzer Zahlen schreiben lässt. Zum Beispiel ist $\frac{21}{13}$ rational, die Zahl $\sqrt{2}$ ist es dagegen nicht. Nun kann man sich fragen, ob es *irrationale* Zahlen $x$ und $y$ gibt, die miteinander potenziert eine *rationale* Zahl als Ergebnis geben. Das folgende Argument zeigt, dass die Antwort darauf positiv ist:

> Zunächst ist die Zahl $\sqrt{2}^{\sqrt{2}}$ entweder rational oder irrational.
>
> - Im ersten Fall sind $x := \sqrt{2}$ und $y := \sqrt{2}$ Zahlen mit der gewünschten Eigenschaft.
> - Im zweiten Fall können wir $x := \sqrt{2}^{\sqrt{2}}$ und $y := \sqrt{2}$ betrachten. Dann sind nämlich $x$ und $y$ irrational und die Potenz $x^y = \sqrt{2}^{\sqrt{2}\cdot\sqrt{2}} = \sqrt{2}^2 = 2$ ist rational.

Mit diesem Beweis kennen wir also die Antwort auf die Frage. Doch halt! Tatsächlich sind wir immer noch nicht in der Lage, einem interessierten Gegenüber ein Zahlenpaar mit den gewünschten Eigenschaften nennen zu können. *Der Beweis war unkonstruktiv.*

In konstruktiver Mathematik stellt man strengere Anforderungen an einen Beweis – so starke, dass man aus jedem konstruktiven Beweis einer Existenzbehauptung einen Algorithmus ablesen kann, der das postulierte Objekt explizit berechnet.

Es stellt sich heraus, dass man, um dieses Ziel zu erreichen, auf genau ein Axiom klassischer Logik verzichten muss: dem *Axiom vom ausgeschlossenen Dritten*. Dieses sagt in etwa aus, dass jede Aussage stimmt oder nicht stimmt; im obigen Beweis ging es gleich im ersten Schritt ein.

Im Vortrag werden wir verstehen, welchen Standpunkt man einnehmen muss, damit der Verzicht auf dieses Axiom nicht völlig verrückt erscheint (ist es nicht einfach offensichtlich wahr?). Außerdem werden wir die *Doppelnegationsübersetzung* kennenlernen, die folgende fundamentale Eigenschaft hat: Genau dann gibt es einen klassischen Beweis einer Aussage – einen, in dem das Axiom vom ausgeschlossenen Dritten verwendet werden darf – wenn es einen konstruktiven Beweis der übersetzten Aussage gibt.

## Die Curry–Howard-Korrespondenz

Die Curry–Howard-Korrespondenz identifiziert *logische Aussagen* mit *Typen* und *konstruktive Beweise von Aussagen* mit *Termen geeigneten Typs*. Unpräzise als Motto formuliert besagt sie, dass Programmieren und konstruktive Mathematik Betreiben ein und dasselbe sind. Unter dieser Korrespondenz entspricht beispielsweise der triviale Beweis der Behauptung „aus $A$ folgt $A$“ der Identitätsfunktion `id :: A → A`. Der triviale Beweis der Behauptung „wenn $A$ und $B$, dann insbesondere $A$“ entspricht der Funktion `fst :: (A,B) → A`. Das werden wir im Vortrag noch genauer thematisieren.

| Logik | Programmierung |
|---|---|
| Aussage $A$ | Typ `A` |
| Konstruktiver Beweis $p$ von $A$ | Term `p :: A` |
| Konjunktion $A \wedge B$ („und“) | Produkttyp `(A,B)` |
| Disjunktion $A \vee B$ („oder“) | Summentyp `Either A B` |
| Implikation $A \Rightarrow B$ („wenn, dann“) | Funktionstyp `A → B` |

Mit der Curry–Howard-Korrespondenz steckt also in jedem konstruktiven Beweis ein Programm; in diesem Sinn hat jeder konstruktive Beweis „algorithmischen Inhalt“. Diese Korrespondenz ist aber auf konstruktive Beweise beschränkt. Nur dank der Doppelnegationsübersetzung kann man auch noch in klassischen Beweisen algorithmischen Inhalt entdecken: indem man die Korrespondenz auf ihre Doppelnegationsübersetzung anwendet.

Welches Pendant auf der Seite der Programmierung gehört eigentlich zur Doppelnegationsübersetzung? *Faszinierenderweise ist das gerade die beim Compilerbau oft verwendete Continuation-Passing-Style-Transformation.* Die CPS-Transformation, mit der man etwa beliebige Kontrollstrukturen recht leicht implementieren kann, wurde in den 60er-Jahren entwickelt. Die Doppelnegationsübersetzung wird seit den 30er-Jahren untersucht. Die Curry–Howard-Korrespondenz vereint also zwei scheinbar unterschiedliche Konzepte aus unterschiedlichen Epochen und unterschiedlichen Teilgebieten der Mathematik und Informatik.

## Fallbeispiel: Minimum einer unendlichen Liste

Ist `[x0, x1, x2, ...]` eine unendliche Liste von natürlichen Zahlen, so ist klar: In dieser Liste ist (mindestens) eines der Elemente das kleinste, das Minimum. Wenn man die Situation genauer untersucht, erkennt man jedoch, dass der Beweis dieser Behauptung das Axiom vom ausgeschlossenen Dritten verwendet und daher nicht konstruktiv ist. In Übereinstimmung mit der Curry–Howard-Korrespondenz können wir auch keine Funktion `minimum :: [Nat] → Nat` schreiben, die das Minimum einer übergebenen unendlichen Liste bestimmen würde.

Was passiert, wenn wir auf den klassischen Beweis die Doppelnegationsübersetzung anwenden, so einen konstruktiven Beweis erhalten und dann diesen mit der Curry–Howard-Korrespondenz in ein Programm umwandeln? Dann erhalten wir eine Funktion vom Typ

```
minimum :: [Nat] → Cont r (Nat, Nat → Cont r ()).
```

Dabei ist `Cont r` die Continuation-Monade. Der Rückgabewert `(n, f)` von `minimum xs` enthält an erster Stelle das Minimum der Liste und an zweiter Stelle eine Funktion, die die Minimalität von `n` bezeugt: Übergibt man ihr einen Folgenindex `i`, so antwortet sie mit einem Beweis der Behauptung `n ≤ xs!!i`. (Um das treu in Haskell abbilden zu können, bräuchte man *abhängige Typen*. Da es die in Haskell nicht gibt, muss der triviale Typ `()` zusammen mit einem sozialen Versprechen als Ersatz herhalten.)

Was macht diese Funktion? *Sie setzt genau das eingangs beschriebene Verfahren um.*

---

Originalliteratur: Chetan Murthy, Extracting Constructive Content from Classical Proofs, 1990. Popularisierungen: in Artikeln von Thierry Coquand, Oleg Kiselyov, Luke Palmer, Dan Piponi (sigfpe), Philip Wadler und Edward Z. Yang. Empfehlenswert ist in diesem Zusammenhang auch ein Buchkapitel von Andrej Bauer.