

---

**Informatik I**
<http://www-pu.informatik.uni-tuebingen.de/info-i-y2k/>


---

## Blatt 9 (alles Bonus)

Abgabe: 11.1.2001

1. [10 Punkte] Auf der Insel der Roboter (siehe unten) fällt Alexander folgendes auf: „Ich habe festgestellt, daß auf einigen der Roboter Ausdrücke stehen, die mit F anfangen und einige, die mit E anfangen. Was hat es damit auf sich?“ „Ah!“ sagte Daniel Chaucey. „Unsere Roboter haben Freundschaften und Feindschaften geformt. Falls x gerade y erzeugt, dann ist Fx ein Freund von y und Ex ist ein Feind von y. Außerdem ist der *beste* Freund von y gerade FCQy und der schlimmste Feind von y ist ECQy. Das hat zu einigen erstaunlichen Phänomenen geführt! Ich zeige Euch was!“
  - (a) „Ein Roboter, der sich selbst als Freund hat, ist *narzistisch*“ sagte Daniel. Es gibt narzistische Roboter. Finde einen! „Außerdem gibt es einen Roboter, der sein eigener Feind ist. Solche Roboter nennen wir *neurotisch*. Da einen narzistischen Roboter gibt, gibt es auch einen neurotischen Roboter.“ Finde einen!
  - (b) Daniel sagt „Es ist natürlich ausgeschlossen, daß ein Roboter sein eigener bester Freund ist.“ Warum?
  - (c) „Aber es gibt einen Roboter, der seinen eigenen besten Freund erzeugt!“ Finde einen!
  - (d) Finde einen Roboter, der seinen schlimmsten Feind zerstört!
2. [10 Punkte] Betrachte folgendes mysteriöse Programm:

```
(define y
  (lambda (f)
    ((lambda (x)
      (f (lambda (z) ((x x) z))))
     (lambda (x)
      (f (lambda (z) ((x x) z))))))))

(define m
  (y
   (lambda (f)
     (lambda (x)
      (if (= x 1)
          1
          (* x (f (- x 1))))))))
```

Was macht m? Y wird auch *Fixpunktkombinator* genannt. Erkläre einem Anfänger, der gerade drei Wochen Informatik I hinter sich hat, diesen Begriff und was Y tut und warum. Gib den dazu passenden Text ab!

3. [10 Punkte] Zur Picture-Sprache gehört die Implementation eines abstrakten Datentyps für Vektoren in  $\mathbb{R}^2$ , bestehend aus den Prozeduren `make-vect`, `vector-xcor`, `vector-ycor`, `vector-add`, `vector-sub` und `vector-scale`.

Des weiteren enthält die Picture-Sprache eine Prozedur `transform-painter`, welche drei Vektoren mit Koordinaten zwischen 0 und 1 als Parameter hat: einen „Ursprung“, eine „Ecke 1“ und eine „Ecke 2“. `transform-painter` liefert eine Prozedur zurück, die einen Painter transformiert.

Außerdem ist da noch eine Prozedur `superpose`, die aus zwei Paintern einen Painter macht, der die Bilder der beiden Parameter übereinanderlegt.

- (a) Welche Repräsentation benutzt die Picture-Bibliothek für Vektoren?
- (b) Finde durch Experimentieren heraus und beschreibe, was `transform-painter` tut.
- (c) Gib unter Benutzung von `transform-painter` und `superpose` selbst-programmierte Versionen von `rotate90`, `beside` und `below` an, sowie Prozeduren, die das Bild eines Painters an der X- sowie der Y-Achse spiegeln.
- (d) Benutze `keep-combining` vom Blatt 4 dazu, um neben `right-split` und `up-split` noch drei weitere Kombinatoren für Painter zu erzeugen, die zu interessanten Bildern führen. Gib jeweils ein auch ein Programm an, mit denen ein solches interessantes Bild erzeugen läßt.

### Die Roboter von Charles Roberts

Jeder Roboter ist mit einer Buchstabenfolge beschriftet. Diese Buchstabenfolge ist ein Programm, das bestimmt, was der Roboter tun wird.

Zunächst zur Terminologie und Notation: Ein *Ausdruck* bzw. eine *Kombination* ist eine Folge von Großbuchstaben — zum Beispiel ist MLBP ein Ausdruck, ebenso LLAZLBA, oder auch nur der einzelne Buchstabe B. Kleinbuchstaben x, y, z, a, b, c stehen für bestimmte Folgen aus Großbuchstaben., und xy steht für die Kombination x gefolgt von der Kombination y. Wenn also zum Beispiel x der Ausdruck MBP ist und y der Ausdruck SLPG, dann ist xy der Ausdruck MBPSLPG. Auch ist Ax dann AMBP, xA ist MBPA, GxH ist GMBPH, CxLy ist CMBPLSLPQ.

Das Programmiersystem für die Roboter basiert auf der Vorstellung, daß bestimmte Ausdrücke die *Namen* anderer Ausdrücke sind. Es gibt zwei Regeln, die das Benennen von Ausdrücken betreffen:

**Regel Q** Für einen Ausdruck x ist der Ausdruck Qx ein Name von x.

Zum Beispiel benennt QBAF gerade BAF, QQH ist ein Name von QH und QDCD benennt DCD.

**Regel R** Wenn x ein Name für y ist, dann ist Rx ein Name für yy.

Also ist zum Beispiel RQB ein Name für BB, da QB ein Name für B ist. Außerdem ist RQBR ein Name für BRBR, da QBR ein Name für BR ist.

Des weiteren gibt es zwei Regeln, die das Erzeugen und Zerstören von Robotern betreffen:

**Regel C** Wenn x ein Name für y ist, so erzeugt Cx gerade y.

Für einen Roboter y erzeugt also der Roboter CQy gerade y.

**Regel D** Falls x y erzeugt, so zerstört Dx y.

Beispielsweise zerstört also DCQx gerade x, da CQx x erzeugt. DCRQx zerstört xx, da CRQx xx erzeugt.