11. Übungsblatt zu
Einführung in die Computationale Logik, SS 2003

Prof. Dr. Gert Smolka, Marco Kuhlmann, MSc
`http://www.ps.uni-sb.de/courses/cl-ss03/`

Lesen Sie im Skript: Kapitel 9

Aufgabe 11.1: Abzählbarkeit

(a) Sei M eine nichtleere Menge. Beweisen Sie, dass M genau dann abzählbar ist, wenn eine surjektive Funktion $\mathbb{N} \to M$ existiert.

(b) Beweisen Sie, dass $\mathbb{N}^3$ abzählbar ist.

(c) Sei M abzählbar und X eine beliebige Menge. Ist $M - X$ abzählbar? Begründen Sie Ihre Antwort.

(d) Sei M abzählbar und X eine endliche Menge. Ist $M \cup X$ abzählbar? Begründen Sie Ihre Antwort.

(e) Zeigen Sie, dass $\mathcal{P}(\mathbb{N})$ überabzählbar ist.

(f) Sei $\mathcal{A} = \{ A \subseteq \mathbb{N} \mid A \text{ abzählbar} \}$. Zeigen Sie, dass $\mathcal{A}$ überabzählbar ist.

Aufgabe 11.2: Fragen zur Berechenbarkeit Ein Kommando heißt *Entscheider* für eine Menge $M \subseteq \mathbb{Z}$, wenn es die Funktion $\lambda x \in \mathbb{Z}.\ \text{if } x \in M \text{ then } 1 \text{ else } 0$ berechnet.

(a) Geben Sie einen Entscheider für $\emptyset$ an.

(b) Geben Sie einen Entscheider für $\mathbb{N}$ an.

(c) Geben Sie einen Entscheider für $\{ x \in \mathbb{Z} \mid -2 \le X \le 13 \}$ an.

(d) Gibt es endliche Teilmengen von $\mathbb{Z}$, die unentscheidbar sind?

(e) Ist die Paarungsfunktion $\mathsf{pair} \in \mathbb{Z} \times \mathbb{Z} \to \mathbb{N}^+$ injektiv? Ist sie surjektiv?

(f) Geben Sie eine Menge $M \subseteq \mathbb{Z}$ an, sodass M nicht prüfbar und $\overline{M}$ prüfbar ist.

(g) Geben Sie eine Menge $M \subseteq \mathbb{Z}$ an, sodass weder M noch $\overline{M}$ prüfbar sind.

(h) Ist die Menge $\{ f \in \mathbb{Z} \to \mathbb{Z}_\perp \mid f \text{ berechenbar} \}$ abzählbar?

(i) Wieviel nicht abzählbare Teilmengen von $\mathbb{Z}$ gibt es?

Aufgabe 11.3: Halteproblem Sei $M = \{ \#c \mid c \in \mathit{Com} \wedge \mathcal{F}(c)(7) \neq 0 \}$. Zeigen Sie, dass M nicht prüfbar ist. Gehen Sie dabei analog zu Satz 9.5.2 vor.

Aufgabe 11.4: Prüfbarkeit Ist die Menge $M = \{\#c \mid c \in Com \wedge \forall x \in \mathbb{Z}\colon \mathcal{F}(c)x = \bot\}$ prüfbar? Beweisen Sie die Richtigkeit Ihrer Antwort.

Aufgabe 11.5: Prüfbarkeit Ist die Menge $M = \{\#c \mid c \in Com \wedge \exists x \in \mathbb{Z}\colon \mathcal{F}(c)x = \bot\}$ prüfbar? Beweisen Sie die Richtigkeit Ihrer Antwort.

IML

Programmieren in IMP ist äußerst mühsam. Wir betrachten daher eine idealisierte Version von SML, die wir IML nennen. IML kann mit beliebig großen ganzen Zahlen rechnen. IML hat keine Referenzen, Ausnahmen oder Strukturen. Mit Standardtechniken aus dem Übersetzerbau kann man in IML geschriebene Prozeduren des Typs `int->int` in Kommandos von IMP übersetzen, die dieselben Funktionen wie die Prozeduren berechnen. Das gilt auch für Prozeduren des Typs `int->bool` und `int->unit`, wenn wir die folgenden Korrespondenzen annehmen:

$$\texttt{true} \mapsto 1, \quad \texttt{false} \mapsto 0, \quad \texttt{()} \mapsto 1$$

Eine Prozedur p vom Typ `int` → `bool` heißt *Entscheider* für eine Menge $M \subseteq \mathbb{Z}$ genau dann, wenn p für alle Argumente terminiert und genau dann `true` liefert, wenn das Argument in M ist. Eine Prozedur p vom Typ `int` → `unit` heißt *Prüfer* für eine Menge $M \subseteq \mathbb{Z}$ genau dann, wenn p für $x \in \mathbb{Z}$ genau dann terminiert, wenn $x \in M$.

Aufgabe 11.6: Gödelisierung in IML

(a) Schreiben Sie in IML Prozeduren für die Gödelisierung von Paaren:

```
pair   : int * int -> int
first  : int -> int
second : int -> int
ispair : int -> bool
```

(b) Schreiben Sie einen Entscheider `aus` für $\#Aus$.

$$Aus \stackrel{\text{def}}{=} \mathbb{Z} \uplus \mathbb{N} \uplus Aus \times Aus$$
$$\#Aus \stackrel{\text{def}}{=} \{\#a \mid a \in Aus\}$$

Aufgabe 11.7: Konstruktion von Prüfern in IML Sie sollen den konstruktiven Teil der Beweise von Satz 9.5.2 und Korollar 9.6.2 in IML programmieren. Nehmen Sie dabei an, dass Prozeduren

```
com  : int -> bool
assn : int -> bool
zuw  : int -> int
seq  : int * int -> int
neg  : int -> int
all  : int -> int
```

gegeben sind, für die gilt:

- `com` entscheidet $\#Com$.
- `assn` entscheidet $\#Assn$.
- $\forall x \in \mathbb{Z}$: $\texttt{zuw}(x) = \#(X_0 := x)$
- $\forall c_1, c_2 \in Com$: $\texttt{seq}(\#c_1, \#c_2) = \#(c_1\,;c_2)$
- $\forall A \in Assn$: $\texttt{neg}(\#A) = \#(\neg A)$
- $\forall A \in Assn$: $\texttt{all}(\#A) = \#(\forall A)$

(a) Schreiben Sie eine Prozedur

```
f : (int -> unit) -> int -> unit
```

die zu einem Prüfer für S_0 einen Prüfer für S liefert.

(b) Schreiben Sie eine Prozedur

```
g : (int -> unit) -> int -> unit
```

die zu einem Prüfer für $\overline{G}$ einen Prüfer für G liefert.