**Algorithmen und Berechnungskomplexität II** SS 16

Universität Bonn, Institut für Informatik, Abteilung I

10. Aufgabenblatt zur Vorlesung **Abgabe: -**

- *Dieses Übungsblatt wird nicht korrigiert und muss deshalb nicht abgegeben werden.*
- *In den Tutorien können Sie Ihre Lösung mit den Erwartungen an die Beweisskizzen abgleichen.*

**Aufgabe 40: Beweis skizzieren (Präsenzaufgabe)**

Skizzieren Sie den Beweis, mit welchem in der Vorlesung gezeigt wurde, dass jede $\mu$-rekursive Funktion auch turingberechenbar ist. ($\mathcal{F}_\mu \subseteq \mathcal{F}_{\text{TM}}$)

*Hinweis: Die Programme der Turingmaschinen müssen nicht explizit angegeben werden. Eine kurze Beschreibung der Idee ist ausreichend.*

**Aufgabe 41: Beweis skizzieren (Präsenzaufgabe)**

Skizzieren Sie den Beweis, mit welchem in der Vorlesung gezeigt wurde, dass jede turingberechenbare Funktion auch $\mu$-rekursiv ist. ($\mathcal{F}_\mu \supseteq \mathcal{F}_{\text{TM}}$)
Gehen Sie dabei insbesondere auf das Diagramm aus Abbildung 2.4 und die Darstellung der $\mu$-rekursiven Funktion am Ende des Beweises ein.

*Hinweis:* Die Beweise von Lemma 9, 10 und 11 müssen nicht skizziert werden.

**Lösung für Aufgabe 40:**

Erwartungshorizont:

- Gegeben sei eine beliebige r-stellige $\mu$-rekursive Funktion $f : \mathbb{D} \to \mathbb{N}_0$ mit $\mathbb{D} \subseteq \mathbb{N}_0^r$. (Das heißt $f$ ist partiell oder total.)
- Zu zeigen ist, dass es eine k-Band DTM M gibt, die $f$ berechnet. D.h. für jedes $\mathfrak{x} = (x_1, x_2, \ldots x_r) \in \mathbb{N}_0^r$ gilt:
  M angesetzt auf die Eingabe
  $$\$\,\text{bin}(x_1)\#\,\text{bin}(x_2)\#\ldots\#\,\text{bin}(x_r)\sqcup$$
  hält genau dann mit der Ausgabe
  $$\$\,\text{bin}(f(\mathfrak{x}))\sqcup$$
  auf dem Band an, wenn $f(\mathfrak{x})$ definiert ist.
- Jede $\mu$-rekursive Funktion kann einer beliebigen Kombination von Grundfunktionen, Operationen und der Anwendung des $\mu$-Operators gebildet werden. Im Folgenden wird kurz gezeigt, wie diese Funktionen bzw. Operationen auf einer DTM realisiert werden können:
  - Konstante Funktionen: $c_s^r$
    Die DTM löscht das Band (auf dem die r-stellige Eingabe steht) und schreibt dann die $\$\,\text{bin}(s)\sqcup$ von links nach rechts aufs Band. Die zu schreibende Ziffernfolge kann über eine Folge von Zuständen abgebildet werden.
  - Nachfolgefunktion: $N$
    Die DTM verschiebt (kopieren + gleichzeitig löschen) $\text{bin}(x)$ von Band 1 auf Band 2. Von links nach rechts wird nun auf Band 2 die Binärzahl um 1 inkrementiert. Dabei wird der Übertrag in einem Zustand gespeichert. Gibt es am Schluss einen Übertrag, so wird dieser im ersten Bandquadrat von Band 1 geschrieben. Anschließend wird der Rest der Addition von Band 2 auf Band 1 kopiert und dabei ggf. an den Übertrag angehängt.
  - Projektionen: $P_i^r$
    Da das $i$ fest ist, kann mit (mehreren) Zuständen der Beginn der $i$-ten Stelle der Eingabe, also der Beginn von $\#\,\text{bin}(x_i)$ gesucht werden. Nun wird $\text{bin}(x_i)$ auf Band 2 gesichert und Band 1 gelöscht.

Anschließend wird der Inhalt von Band 2 wieder auf Band 1 verschoben.

  - Substitution: $g(g_1(\mathfrak{x}), g_2(\mathfrak{x}), \dots g_m(\mathfrak{x}))$
    Die Maschine verwendet 2 zusätzliche Bänder. Eines davon, um die Eingabe $\mathfrak{x}$ zu dauerhaft zu sichern und für die Berechnungen der DTMs der $g_i$ wiederzuverwenden. Das andere, um die nacheinander berechneten Ergebnisse der $M_i$ zu sammeln und zu verketten. Diese Kette von Ergebnissen bildet dann die Eingabe für die DTM $M_g$. Nun kann $M_g$ mit dieser Eingabe gestartet werden.
  - Primitive Rekursion: $h(\mathfrak{x}, 0) = g(\mathfrak{x}), h(\mathfrak{x}, n+1) = f(\mathfrak{x}, n, h(\mathfrak{x}, n))$
    Die DTM verwendet 4 zusätzliche Bänder. Eines, um die Eingabe $\mathfrak{x}$ dauerhaft zu speichern. Das zweite, um berechnete Zwischenergebnisse zwischenzuspeichern. Die beiden anderen dienen als Zähler für die Anzahl der noch durchzuführenden bzw. bereits durchgeführten Rekursionsdurchläufe. Nun werden die DTMs (erst $M_g$, dann mehrmals $M_f$) ausgeführt und dabei die Zähler und das Zwischenergebnis angepasst. Ist der Zähler für die noch durchzuführenden Schritte 0, wird das Ergebnis auf Band 1 geschrieben.
  - $\mu$-Operator: $\mu f$
    Die DTM verwendet 3 zusätzliche Bänder. Eines dient zum dauerhaften speichern der Eingabe $\mathfrak{x}$. Das zweite dient als Zähler von 0 aufwärts bis $n$, falls es existiert. Auf dem dritten Band wird zu Beginn 1 und dann stets das Ergebnis der letzten Berechnung von $M_f$ geschrieben. Ist dieses Band 0, so wurde die Nullstelle gefunden und der Zählerstand ausgegeben.

- Da jede Grundfunktion durch eine DTM berechnet werden kann und jede der Operationen und der $\mu$-Operator wie beschrieben auf die DTM (und damit die von ihr berechneten Funktion) angewendet werden kann, können wir auch für $f$ eine DTM konstruieren. Damit ist $f$ nach Definition turingberechenbar.

**Lösung für Aufgabe 41:**

- Für den Beweis konnten wir uns auf 1-Band DTMs beschränken. Dies ist möglich, da jede k-Band DTM durch eine 1-Band DTM simuliert werden kann. Dadurch kann der Zustand einer 1-Band DTM als Konfiguration K dargestellt werden:

$$\$a_1 \ldots a_{k-1}\ q\ a_k \ldots a_t$$

Hierbei entspricht $a_i \in \Sigma$ dem Inhalt des i-ten Bandquadrates, $q$ dem Zustand und $a_k$ der Kopfposition.

- Das folgende Diagramm verdeutlicht, wie der Wechsel einer DTM von einer Konfiguration K in die Folgekonfiguration K' durch eine primitiv rekursive Funktionen berechnen werden kann.

$$\begin{tikzcd}
K \arrow[r, mapsto, "\Delta"] \arrow[d, "\Psi"'] & K' \arrow[d, "\Psi"] \\
\Psi(K) \arrow[r, mapsto, "\tilde{\Delta}"] & \Psi(K')
\end{tikzcd}$$

Die Bestandteile des Diagramms sind

K: Eine beliebige Konfiguration (als String) der DTM.

K': Die Folgekonfiguration der DTM, die sich durch Anwenden der Übergangsfunktion $\Delta$ der DTM ergibt.

$\Psi$: Mit dieser Funktion können beliebige Konfiguration injektiv auf die natürlichen Zahl abgebildet werden.

$\tilde{\Delta}$: Diese primitiv rekursive Funktion macht das Diagramm kommutativ. Sie gibt die Kodierung $\Psi(K')$ der Folgekonfiguration von $\Psi(K)$ zurück, falls K eine korrekte Konfiguration der DTM ist. Andernfalls soll die Funktion 0 zurückgeben. Sie führt sozusagen den Konfigurationsübergang auf den natürlichen Zahlen durch.

$\tilde{\Delta}$ kann mit Hilfe primitiv rekursiver Stringfunktionen konstruiert werden. Dazu wird zunächst die Kodierung einer Konfiguration in den Konfigurationsstring mit $\Psi^{-1}$ umgerechnet. Der String wird dann in drei Teile zerlegt, der durch die DTM gegebene Übergang lokal durchgeführt und die Teile wieder zum String der Folgekonfiguration zusammengesetzt. Durch Anwenden von $\Psi$ kann die Folgekonfiguration dann in die Folgekodierung umgewandelt werden.

- Dass zu jeder turingberechenbaren Funktion schließlich eine $\mu$-rekursive Funktion angegeben werden kann zeigt die folgende Formel.

$$f(x_1, \ldots x_r) = F\Bigg(D\Big(A(E(x_1, \ldots, x_r)), E(x_1, \ldots, x_r)\Big)\Bigg).$$

  Dabei erfüllen die verwendeten Funktionen folgende Aufgaben:

  E: Diese primitiv rekursive Funktion dient zur Kodierung der Startkonfiguration, d.h. zur Berechnung der Kodierung des Strings $\$q_0\,\mathrm{bin}(x_1)\#\,\mathrm{bin}(x_2)\#\ldots\#\,\mathrm{bin}(x_r)$ mit der Abbildung $\Psi$.

  A: Diese Funktion sucht in der Folge der kodierten Folgekonfigurationen der Startkonfiguration nach einer Endkonfiguration. Für diese Suche ist der $\mu$-Operator notwendig. Die DTM muss nicht notwendigerweise halten, in diesem Fall ist $f$ partiell. Wird jedoch eine Endkonfiguration gefunden, so gibt $A$ die (minimale) Anzahl an Rechenschritten zurück, die die DTM benötigt, um von der Startkonfiguration in die Endkonfiguration zu gelangen.

  D Diese primitiv rekursive Funktion führt auf der Kodierung der Startkonfiguration solange einen Konfigurationsübergang lokal mit $\tilde{\Delta}$ durch, wie von der Funktion A berechnet wurde. Dadurch befindet sich die DTM nach allen Rekursionsschritten in einer Endkonfiguration, deren Kodierung zurückgegeben wird.

  F: Diese primitiv rekursive Funktion berechnet aus der Kodierung der Endkonfiguration zunächst den Konfigurationsstring und daraus das Ergebnis der Berechnung als String. Dieser Ergebnis-String $\$\,\mathrm{bin}(x)$ wird dann von F in das Ergebnis von $f$ umgewandelt.