RWTH Aachen **WS 2012/13**
Lehrgebiet Theoretische Informatik Übungsblatt 3
Reidl–Ries–Rossmanith–Sanchez–Tönnis 29.10.2012

Übung zur Vorlesung Berechenbarkeit und Komplexität

Aufgabe T6

1. Zeigen Sie, daß jede RAM durch eine RAM mit einer festen Anzahl von Registern simuliert werden kann.

Hinweis: Als Zwischenschritt können Sie auch eine RAM durch eine Turingmaschine simulieren.

Lösungsvorschlag Wir simulieren erst die RAM mit beliebiger Anzahl der beliebiger Register durch eine TM M, nach VL ist dies möglich. Jetzt muss nur noch gezeigt werden, dass M durch eine RAM mit konstanter Anzahl Register simuliert werden kann. Dazu speichern wir den Zustand in Register 1, dies ist möglich da beliebig große Zahlen in ein Register gespeichert werden können. Den Inhalt des Bandes links von Kopf und rechts von Kopf kann jeweils in Register 2 und 3 gespeichert werden. Der Bandinhalt kann dabei zum beispiel im Tenärsystem gespeichert werden. Zugriffe auf Bandinhalte kann dann über MOD- und DIV operatoren passieren.

2. Wir wollen zeigen, dass der Befehlssatz der RAM für die Simulation auf die Befehle LOAD, CLOAD, STORE, CADD, CSUB, GOTO, IF $c(0) \neq 0$ GOTO und END eingeschränkt werden kann. Hierzu nutzen wir aus, daß wir nur eine RAM mit konstant vielen Registern simulieren müssen.

Zeigen Sie, wie die Befehle ADD i und INDLOAD i durch den eingeschränkten Befehlssatz ersetzt werden können.

- ADD i

  Lösungsvorschlag Der Befehl ADD i bewirkt ein $c(0) = c(0) + c(i)$. Da der Akkumulator an den meisten Befehlen implizit beteiligt ist, und wir das Register i nicht verändern wollen, benutzen wir zwei Hilfsregister $c(k+1) = c(0)$ und $c(k+2) = c(i)$. Etwas abstrahiert und in einer Hochsprache sieht der Befehl dann wie folgt aus.

```
c(k+1)=c(0);
c(k+2)=c(i);
while (c(k+2)>0)
{
  c(k+1)=c(k+1)+1;
  c(k+2)=c(k+2)-1;
}
c(0)=c(k+1);
```

Dabei wird davon ausgegangen, dass das ursprüngliche RAM-Programm nur die Register $c(0)$ bis $c(k)$ benutzt. Übersetzt lautet der RAM-Befehl dann:

```
  STORE k+1
  LOAD i
  STORE k+2
  IF c(0)!=0 GOTO while
  GOTO end
$while
  LOAD k+1
  CADD 1
  STORE k+1
  LOAD k+2
  CSUB 1
  STORE k+2
  IF c(0)!=0 GOTO while
$end
  LOAD k+1
```

- INDLOAD i

  Lösungsvorschlag

```
  LOAD i;
  IF c(0)!=0 GOTO biggerthan0
  LOAD 0
  GOTO end
$biggerthan0
  CSUB 1
  IF c(0)!=0 GOTO biggerthan1
  LOAD 1
  GOTO end
  ...
$biggerthan(k-2)
  CSUB 1
  IF c(0)!=0 GOTO biggerthan(k-1)
  LOAD k-1
  GOTO end
$biggerthan(k-1)
  LOAD k
$end
```

Aufgabe T7

Zeigen sie mithilfe der Unterprogrammtechnik, daß die Sprachen

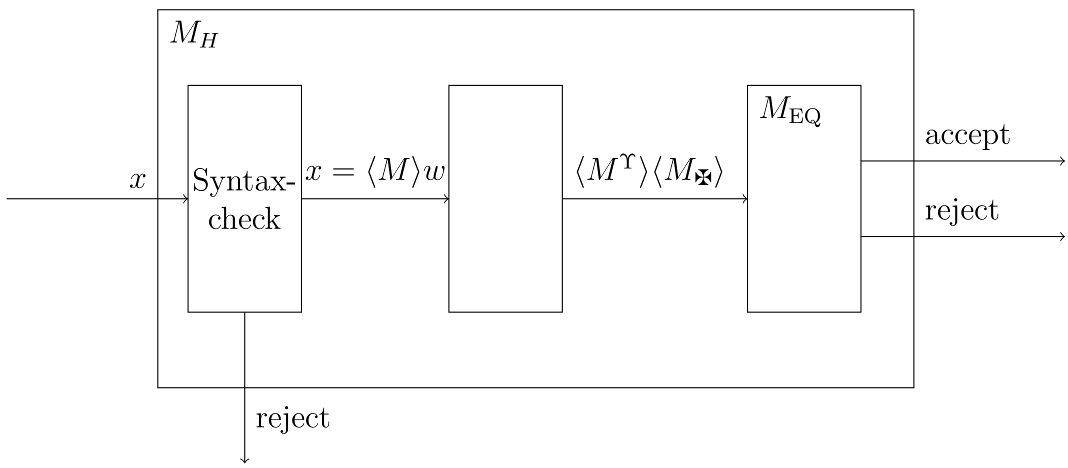
1. $A = \{\langle M\rangle w \mid M \text{ akzeptiert } w\}$.
2. $A_{\text{EQ}} = \{\langle M_1\rangle\langle M_2\rangle \mid L(M_1) = L(M_2)\}$.

nicht entscheidbar sind. Verwenden Sie *nicht* den Satz von Rice.

Lösungsvorschlag

1.
   - Zum Widerspruch nehmen wir an, es gibt eine TM M_A, die die Sprache A entscheidet.
   - Gemäß der Definition *rekursiver Sprachen* hält M_A auf jeder Eingabe x und akzeptiert genau dann, wenn $x = \langle M \rangle w \in A$.
   - Wir konstruieren nun eine TM $M_{\bar{D}}$ für das Komplement der Diagonalsprache, die M_A als Unterprogramm verwendet und $\bar{D}$ entscheidet: Zu einer Eingabe x berechnet $M_{\bar{D}}$ zunächst denjenigen Index i, so dass $x = w_i$. Dann berechnet $M_{\bar{D}}$ die Gödelnummer $\langle M_i \rangle$ der Maschine M_i und ruft die Maschine M_A mit der Eingabe $\langle M_i \rangle w_i$ auf. Anschließend übernimmt sie deren Entscheidung.
   - Die TM $M_{\bar{D}}$ entscheidet nun offensichtlich $\bar{D}$. Ein Widerspruch zur Unentscheidbarkeit von $\bar{D}$.

2.
   - Zum Widerspruch nehmen wir an, es gibt eine TM M_{EQ}, die die Sprache A_{EQ} entscheidet.
   - Gemäß der Definition *rekursiver Sprachen* hält M_{EQ} auf jeder Eingabe x und akzeptiert genau dann, wenn $x = \langle M_1 \rangle \langle M_2 \rangle \in A_{\mathrm{EQ}}$.
   - Wir konstruieren nun eine TM M_H für das Halteproblem, die M_{EQ} als Unterprogramm verwendet und H entscheidet: Zu einer Eingabe x prüft M_H zunächst, ob $x = \langle M \rangle w$ gilt und die Eingabe somit die richtige Form hat. Ist dies nicht der Fall, so verwirft M_H.
   - Nun berechnet M_H die Gödelnummern zweier Turingmaschinen $\langle M^{\Upsilon} \rangle$ und $\langle M_{\maltese} \rangle$ mit den nachfolgenden Eigenschaften:
   - M^{Υ} löscht zunächst ihre Eingabe und schreibt dann w auf das Eingabeband. Dann verhält sich M^{Υ} genau wie M, bis M hält. Wenn M hält, so akzeptiert M^{Υ}.
   - $M_{\maltese}$ ist eine TM, die auf allen Eingaben hält und die immer akzeptiert.
   - Nun ruft die Turingmaschine M_H die TM M_{EQ} mit der Eingabe $\langle M^{\Upsilon} \rangle \langle M_{\maltese} \rangle$ auf und übernimmt das Akzeptanzverhalten.



Offensichtlich terminiert die konstruierte TM M_H auf allen Eingaben.

Korrektheit:

Falls w nicht mit einer gültigen Gödelnummer beginnt, so ist $w \notin H$ und M_H verwirft die Eingabe w.

Sei nun $w = \langle M \rangle w$. Es gilt:

$$\begin{aligned} w \in H &\Rightarrow M \text{ hält auf } w \\ &\Rightarrow \langle M^{\Upsilon} \rangle \text{ akzeptiert } w \\ &\Rightarrow \langle M^{\Upsilon} \rangle \text{ und } \langle M_{\maltese} \rangle \text{ akzeptieren die selben Sprachen} \\ &\Rightarrow \langle M^{\Upsilon} \rangle \langle M_{\maltese} \rangle \in M_{\mathrm{EQ}} \\ &\Rightarrow M_{\mathrm{EQ}} \text{ akzeptiert } \langle M^{\Upsilon} \rangle \langle M_{\maltese} \rangle \\ &\Rightarrow M_H \text{ akzeptiert } w \end{aligned}$$

$$\begin{aligned} w \notin H &\Rightarrow M \text{ hält nicht auf } w \\ &\Rightarrow \langle M^{\Upsilon} \rangle \text{ hält nicht} \\ &\Rightarrow \langle M^{\Upsilon} \rangle \text{ und } \langle M_{\maltese} \rangle \text{ akzeptieren nicht die selben Sprachen} \\ &\Rightarrow \langle M^{\Upsilon} \rangle \langle M_{\maltese} \rangle \notin M_{\mathrm{EQ}} \\ &\Rightarrow M_{\mathrm{EQ}} \text{ verwirft } \langle M^{\Upsilon} \rangle \langle M_{\maltese} \rangle \\ &\Rightarrow M_H \text{ verwirft } w \end{aligned}$$

Die TM M_H entscheidet damit H. Dies ist aber ein Widerspruch zur Unentscheidbarkeit von H. Die getroffene Annahme, dass die TM M_{EQ} existiert, ist somit falsch und die Sprache A_{EQ} ist unentscheidbar.

Aufgabe H5 (5 Punkte)

Führen Sie den zweiten Teil der Aufgabe T6 für den Befehl IF $c(0) < x$ GOTO j durch.

Lösungsvorschlag

Statt nach $c(0) < x$ können wir auch nach $c(0) + 1 - x \leq 0$ fragen. Da x konstant ist, ist diese Umwandlung trivial.

```
  STORE k+1
  CADD 1
  CSUB x
  IF c(0)!=0 GOTO continue
  LOAD k+1
  GOTO j
$continue
  LOAD k+1
```

Aufgabe H6 (3 Punkte)

Beweisen oder widerlegen Sie die folgende Behauptung: Eine RAM, deren Registergröße beschränkt ist (zum Beispiel auf 32 bit) kann jede Turingmaschine simulieren.

Lösungsvorschlag Die Aussage ist falsch, da mit endlich großen Registern auch nur endlich viel Speicher adressiert werden kann.

Aufgabe H7 (9 Punkte)
Zeigen Sie mithilfe der Unterprogrammtechnik, daß die Sprache

$$H_1 = \{\langle M\rangle \mid M \text{ hält auf jeder Eingabe und akzeptiert mindestens ein Wort}\}$$

nicht entscheidbar ist. Gehen Sie dabei besonders auf die Korrektheit ein.

Lösungsvorschlag Um $H_{\text{all}} \leq H_1$ zu zeigen, konstruieren wir eine berechenbare Funktion $f : \Sigma^* \to \Sigma^*$, so dass $x \in H_{\text{all}} \Leftrightarrow f(x) \in H_1$ gilt.
Sei x das Argument von f. Falls x keine gültige Gödelnummer ist, so setze $f(x) = x$. Andernfalls ist $x = \langle M\rangle$ einer TM M. Wir konstruieren nun die Gödelnummer einer TM M' und setzen $f(x) = \langle M'\rangle$. M' arbeitet dabei wie folgt: M' simuliert auf Eingabe w die TM M. Wenn M hält, dann akzeptiert M'.
Offensichtlich ist die Funktion f berechenbar.
Korrektheit:
Falls x keine gültige Gödelnummer ist, so ist $x \notin H_{\text{all}}$ und gemäß der Konstruktion gilt $f(x) \notin H_1$.
Sei nun $x = \langle M\rangle$. Es gilt:

$$\begin{aligned} x \in H_{\text{all}} &\Rightarrow M \text{ hält auf jeder Eingabe} \\ &\Rightarrow M' \text{ hält auf jeder Eingabe und akzeptiert jede Eingabe} \\ &\Rightarrow f(x) = \langle M'\rangle \in H_1 \end{aligned}$$

$$\begin{aligned} x \notin H_{\text{all}} &\Rightarrow M \text{ hält nicht auf jeder Eingabe} \\ &\Rightarrow M' \text{ hält nicht auf jeder Eingabe} \\ &\Rightarrow f(x) = \langle M'\rangle \notin H_1 \end{aligned}$$

Aufgabe H8 (späterer Abgabetermin)
Geben Sie eine Turingmaschine mit fünf Zuständen und Bandalphabet $\Gamma = \{\, 0, 1, B \,\}$ an, die auf der leeren Eingabe ϵ terminiert, so daß nach Ende der Berechnung eine möglichst hohe Anzahl an 1en auf dem Band steht.
Testen Sie ihre Konstruktion mit Ihrem Simulator vom zweiten Übungsblatt. Geben Sie die Anzahl der 1en an, die nach der Terminierung auf dem Band stehen.
Für diese Aufgabe haben Sie Zeit bis zum ersten Termin ihres Tutoriums nach den Weihnachtsferien. (Sie dürfen Ihre Lösung natürlich auch früher abgeben.)

Lösungsvorschlag TODO