Theoretische Grundlagen der Informatik
Tutorium XII & XIII (SR 301)
Tut Nr. 5 – Üb5, Entscheidbarkeit

David Münch

Karlsruher Institut für Technologie
Institut für Informatik
IKS Müller-Quade

26. November 2009

Karlsruher Institut für Technologie

Inhaltsverzeichnis

Inhaltsverzeichnis

Inhaltsverzeichnis

Inhaltsverzeichnis

Organisatorisches

Email: muenchdavid@gmail.com

`https://www.stud.uni-karlsruhe.de/~uhbro/`

Tutorium 12: Donnerstags 8:00 Uhr - Raum 301
Tutorium 13: Donnerstags 9:45 Uhr - Raum 301

Übungsblattabgabe Mittwochs 12:00 Uhr.

Organisatorisches

Deckblatt benutzen: `http://www.stud.uni-karlsruhe.de/~unbdh/deckblatt/index.php?course=6`

Gruppenarbeit erwünscht, aber jeder muss handschriftliche Lösung mit Namen aller Gruppenteilnehmer abgeben.

50% der Punkte sind notwendig für den Schein.

Organisatorisches

Nicht abgeholte Übungsblätter können ab sofort bei Nico Döttling R274 abgeholt werden.

`https://puck.iaks.uka.de/eiss/?id=167`

Was wollen wir heute erreichen?

Was wollen wir heute erreichen?

- Übungsblatt 4 besprechen

Was wollen wir heute erreichen?

- Übungsblatt 4 besprechen
- Wortproblem

Was wollen wir heute erreichen?

- Übungsblatt 4 besprechen
- Wortproblem
- Universelle Turingmaschine

Was wollen wir heute erreichen?

- Übungsblatt 4 besprechen
- Wortproblem
- Universelle Turingmaschine
- Halteproblem

Was wollen wir heute erreichen?

- Übungsblatt 4 besprechen
- Wortproblem
- Universelle Turingmaschine
- Halteproblem
- Postsches Korrespondenzproblem

Aufgabe 1

Gegeben sei die Sprache

$$\mathcal{L} = \{a^i b^j c^k | 0 \leq i \leq j \leq k\} \subseteq \{a, b, c\}^*$$

Zeigen Sie:

a) $\mathcal{L}$ ist nicht kontextfrei.

b) $\mathcal{L}$ ist vom Chomsky-Typ 1. Geben Sie dazu explizit eine linear beschränkte Turingmaschine an die $\mathcal{L}$ erkennt.

Definition: Pumping-Lemma für kontextfreie Sprachen

Für jede kontextfreie Sprache L gibt es eine Konstante $p \in \mathbb{N}$, so dass sich jedes Wort $w \in L$ mit $|w| \geq p$ so als $w = uvxyz$ schreiben lässt, dass $|vy| \geq 1$, $|vxy| \leq p$ und für alle $i \geq 0$ das Wort $uv^ixy^iz \in L$ ist.

Lösung 1 a)

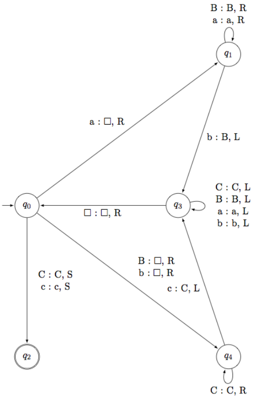
Annahme: $\mathcal{L}$ ist kontextfrei.
Sei p die Konstante aus dem Pumping-Lemma. Wir wählen das Wort $w = a^p b^p c^p \in \mathcal{L}$. Sei $w = uvxyz$ eine Zerlegung des Wortes w mit $|vxy| \leq p$ und $|vy| > 0$. Wir machen folgende Fallunterscheidung:

1. Die Trivialfälle für welche v oder y zwei verschiedene Symbole enthalten führen wir nicht aus, da dann nach einmaligem Pumpschritt das so erhaltene Wort nicht mehr von der Form $a^i b^j c^k$ ist.
2. Befindet sich vxy vollständig in a^p, so liegt $w' = uv^2xy^2z = a^{p+t}b^pc^p \notin \mathcal{L}$ da $t > 0$.
3. Befindet sich v im Teilwort a^p und y in b^p, so liegt $w' = uv^2xy^2z = a^{p+r}b^{p+s}c^p \notin \mathcal{L}$, da mindestens eines von s und $t > 0$ ist.
4. Befindet sich uvx vollständig in b^p, so argumentieren wir analog zu Fall 2.
5. Befindet sich v in b^p und y in c^p, so liegt $w' = uv^0xy^0z = a^pb^{p-s}c^{p-r} \notin \mathcal{L}$, da entweder s oder $r > 0$.
6. Befindet sich vwx vollständig in c^p, so liegt $w' = uv^0xy^0z = a^pb^pc^{p-r} \notin \mathcal{L}$, da $r > 0$

Dies ist ein Widerspruch zum PL $\Rightarrow \mathcal{L}$ ist nicht kontextfrei.

Lösung 1 b)

Sei $\mathcal{M} = (\mathcal{Q}, \Sigma, \Gamma, \delta, q_0, \Box, \{q_2\})$ mit $\mathcal{Q} = \{q_0, \ldots, q_4\}$, $\Sigma = \{a, b, c\}$, $\Gamma = \{a, b, c, B, C, \Box\}$. Die Zustandsübergangsfunktion δ ist gegeben durch:



So kann die Turingmaschine als Programm für `http://ironphoenix.org/tril/tm/` aussehen.

$1, a\ 2, \_, >$
$1, B\ 5, \_, >$
$1, b\ 5, \_, >$
$1, C\ H, C, >$
$1, c\ H, c, >$
$2, B\ 2, B, >$
$2, a\ 2, a, >$
$2, b\ 4, B, <$
$5, C\ 5, C, >$
$5, b\ 5, b, >$
$5, B\ 5, B, >$
$5, c\ 4, C, <$
$4, \_\ 1, \_, >$
$4, C\ 4, C, <$
$4, B\ 4, B, <$
$4, a\ 4, a, <$
$4, b\ 4, b, <$
$1, \_\ H, \_, >$

Aufgabe 2

Gegeben sei die Sprache

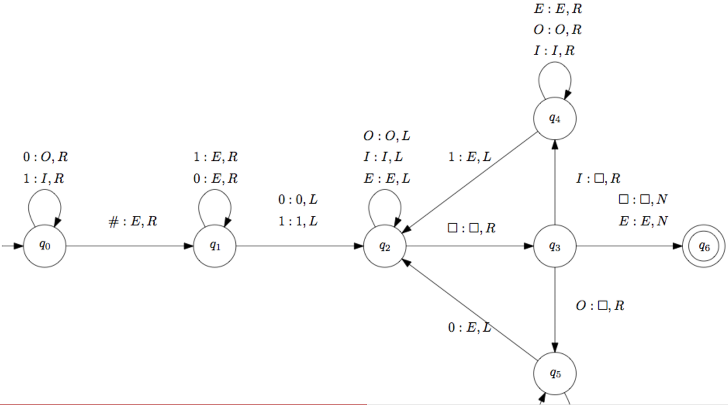
$$\mathcal{L} = \{w\#x | w, x \in \{0,1\}^* \ w \text{ ist Teilwort von } x \ \}$$

a) Geben Sie explzit eine *nichtdeterministische* Turingmaschine $\mathcal{M}$ an, die $\mathcal{L}$ erkennt.

b) Geben sie eine akzeptierende Konfigurationenabfolge ihrer Maschine $\mathcal{M}$ für das Eingabewort $0\#1011$ an.

Lösung 2 a)

Sei $\mathcal{M} = (\mathcal{Q}, \Sigma, \Gamma, \delta, q_0, \Box, \{q_6\})$ mit $\mathcal{Q} = \{q_0, \ldots, q_6\}$, $\Sigma = \{0, 1, \#\}$, $\Gamma = \{0, 1, \#, I, O, E, \Box\}$. Der Zustandsübergangsgraph für δ steht nachfolgend.



Lösung 2 b)

$$(q_0)0|\#|1|0|1|1$$
$$O|(q_0)\#|1|0|1|1$$
$$O|E|(q_1)1|0|1|1$$
$$O|E|E|(q_1)0|1|1$$
$$O|E|(q_2)E|0|1|1$$
$$O|(q_2)E|E|0|1|1$$
$$(q_2)O|E|E|0|1|1$$
$$(q_1)\square|O|E|E|0|1|1$$
$$(q_3)O|E|E|0|1|1$$

$$(q_5)E|E|0|1|1$$
$$E|(q_5)E|0|1|1$$
$$E|E|(q_5)0|1|1$$
$$E|(q_2)E|E|1|1$$
$$(q_2)E|E|E|1|1$$
$$(q_2)\square|E|E|E|1|1$$
$$(q_3)E|E|E|1|1$$
$$(q_6)E|E|E|1|1$$

Als Wortproblem einer formalen Sprache bezeichnet man das Entscheidungsproblem, zu einem gegebenen Wort festzustellen, ob dieses zur Sprache gehört oder nicht.
Das Wortproblem einer Sprache L ist entscheidbar, wenn ihre charakteristische Funktion χ_L berechenbar ist. Sie ist definiert durch

$$\chi_L : \Sigma^* \to \{0,1\};\ w \mapsto \begin{cases} 1, & \text{falls } w \in L \\ 0, & \text{sonst} \end{cases}$$

Die Sprache L hat also ein entscheidbares Wortproblem, wenn es einen Algorithmus gibt, der in endlicher Zeit herausfindet, ob $w \in L$ oder nicht. Jedes Entscheidungsproblem lässt sich als Wortproblem einer formalen Sprache codieren.

CH-Typ	entscheidbar	Laufzeit
3	Ja, endlicher Automat	$O(n)$
2	Ja, CYK	$O(n^3)$
1	Ja, Algorithm 1	$\lvert\Sigma\rvert^{O(n)}$
0	Nein, RA	NP-hart

Algorithm 1 WORTPROBLEM FÜR KONTEXTSENSITIVE SPRACHEN

input : Grammatik $G = (T, V, S, P)$
output: $w \in L(G)$
$M = \emptyset$, $M' = \{S\}$
while $M \neq M'$ **do**
  $M = M'$
  $M' = M \cup \{\beta | \alpha \to \beta, \alpha \in M$ und $|\beta| \leq |w|\}$
end
if $w \in M$ **then**
  $w \in L(G)$
else
  $w \notin L(G)$
end

Aufgabe

Gegeben sei folgende Grammatik
$\mathcal{G} = (\{a, b, c\}, \{S, A, B, C\}, S, \mathcal{P})$ mit
$\mathcal{P} = \{S \to aABcb \mid aBC \mid CBABc,$
$aA \to BCaAa \mid bb,$
$BC \to CB,$
$B \to aCaa \mid babAcc,$
$aCa \to aAc \mid aca\}.$

Lösen Sie das Wortproblem für das Wort $w = acbbca$!

Lösung:

Mit $|\omega| = 6$ ergibt die Anwendung des Algorithmus für kontextfreie Grammatiken folgende Tabelle:

Durchlauf	$\mathcal{M}'$
0	$\mathcal{M}_0 := \{S\}$
1	$\mathcal{M}_1 := \mathcal{M}_0 \cup \{aABcb, aBC, CBABc\}$
2	$\mathcal{M}_2 := \mathcal{M}_1 \cup \{bbBcb, aCB, aaCaaC\}$
3	$\mathcal{M}_3 := \mathcal{M}_2 \cup \{aCaCaa, aaAcaC, aacaaC\}$
4	$\mathcal{M}_4 := \mathcal{M}_3 \cup \{abbcaC, aAcCaa, acaCaa, aCaAca, aCacaa$
5	$\mathcal{M}_5 := \mathcal{M}_4 \cup \{bbcCaa, aCbbca, acaAca, acacaa, aAcAca, a$
6	$\mathcal{M}_6 := \mathcal{M}_5 \cup \{acbbca, \ldots\}$ bzw. $\mathcal{M}_6 := \mathcal{M}_5 \cup \{acbbca, bbc$
7	$\mathcal{M}_7 := \mathcal{M}_6 \cup \varnothing$

$acbbca \in \mathcal{M}_6 \Rightarrow acbbca \in \mathcal{L}(\mathcal{G})$

Universelle Turingmaschine $\mathcal{U}$:
Nimmt als Eingabe eine $0, 1$-Codierung einer Turingmaschine $\mathcal{M}$ und eine Eingabe $k \in \{0, 1\}^*$. $\mathcal{U}$ simuliert dann die Berechnung von $\mathcal{M}$ bei Eingabe k und akzeptiert genau dann wenn $\mathcal{M}$ akzeptiert.

Eine Codierung einer Turingmaschine über einem endlichen Alphabet Σ oder äquivalent dazu N nennt sich Gödelisierung (nach Kurt Gödel). Für eine gegebene Turingmaschine $\mathcal{M}$ schreiben wir $\langle \mathcal{M} \rangle$ für die Gödelnummer von $\mathcal{M}$. Für gegebene Gödelnummer w schreiben wir $\mathcal{M}_w$ für die zugehörige Turingmaschine.

Definition:Gödelnummer einer Turingmaschine

oBdA. immer $\mathcal{M}_w = (\{q_1, ..., q_t\}, \{0,1\}, \{0,1,\sqcup\}, \delta, q_0, \{q_t\})$
Definiere immer nur δ:
$\delta(q_i, X_j) = (q_k, X_l, D_m)$
$code_t = 0^i 1 0^j 1 0^k 1 0^l 1 0^m$
Dann ist $w = 111 code_1 11 code_2 11 ... 11 code_s 111 \in \mathbb{N}$ die Gödelnummer von $\mathcal{M}_w$.
Auch: $\langle \mathcal{M} \rangle$ statt w.

Man kann sich eine Gödelnummer als eine Formalisierung von Computerprogrammen vorstellen.

Die universelle Turingmaschine ist dann der Interpreter der dieses Programm ausführt (Perl, Python etc.).

Definition:

- Eine Sprache $L \subseteq \Sigma^*$ heisst **rekursiv** oder **entscheidbar**, wenn es eine TM gibt, die auf allen Eingaben stoppt und eine Eingabe w genau dann akzeptiert, wenn $w \in L$ gilt.

Definition:

- Eine Sprache $L \subseteq \Sigma^*$ heisst **rekursiv** oder **entscheidbar**, wenn es eine TM gibt, die auf allen Eingaben stoppt und eine Eingabe w genau dann akzeptiert, wenn $w \in L$ gilt.
- Eine Sprache $L \subseteq \Sigma^*$ heisst **rekursiv-aufzählbar** oder **semi-entscheidbar**, wenn es eine TM gibt, die genau die Eingaben w akzeptiert für die $w \in L$. (Das Verhalten der TM für $w \notin L$ ist nicht defniniert.)

Definition:

- Eine Funktion $f : \Sigma^* \to \Gamma^*$ heisst **berechenbar** oder **totalrekursiv**, wenn es eine TM gibt, die bei Eingabe von $w \in \Sigma^*$ den Funktionswert $f(w) \in \Gamma^*$ ausgibt.

Definition:

- Eine Funktion $f : \Sigma^* \to \Gamma^*$ heisst **berechenbar** oder **totalrekursiv**, wenn es eine TM gibt, die bei Eingabe von $w \in \Sigma^*$ den Funktionswert $f(w) \in \Gamma^*$ ausgibt.
- Eine TM realisiert eine Funktion $f : \Sigma^* \to \Gamma^*$, falls gilt:

$$f(w) = \begin{cases} \text{Ausgabe der TM}, & \text{wenn sie bei Eingabe } w \text{ stoppt.} \\ \text{undefiniert}, & \text{sonst} \end{cases}$$

Definition:

- Eine Sprache $L \subseteq \Sigma^*$ ist **entscheidbar** $\Leftrightarrow$ ihre charakterische Funktion χ_L berechenbar ist, wobei gilt:

$$\chi_L : \Sigma^* \to \{0,1\} \text{ mit } \chi_L(w) = \begin{cases} 1, & \text{falls } w \in L \\ 0, & \text{sonst.} \end{cases}$$

Definition:

- Eine Sprache $L \subseteq \Sigma^*$ ist **entscheidbar** $\Leftrightarrow$ ihre charakterische Funktion χ_L berechenbar ist, wobei gilt:

$$\chi_L : \Sigma^* \to \{0,1\} \text{ mit } \chi_L(w) = \begin{cases} 1, & \text{falls } w \in L \\ 0, & \text{sonst.} \end{cases}$$

- Eine Sprache $L \subseteq \Sigma^*$ ist **semi-entscheidbar** $\Leftrightarrow$ Funktion χ_L^* berechenbar ist, wobei gilt:

$$\chi_L^*(w) = \begin{cases} 1, & \text{falls } w \in L \\ \text{undefiniert}, & \text{sonst.} \end{cases}$$

Definition: Many-One Reduzierbar

Eine Sprache $\mathcal{A}$ ist many-one reduzierbar auf eine Sprache $\mathcal{B}$, kurz $\mathcal{A} \leq_m \mathcal{B}$, falls eine berechenbare Funktion $f : \Sigma^* \to \Sigma^*$ existiert, sodass für alle $w \in \Sigma^*$

$$w \in \mathcal{A} \Leftrightarrow f(w) \in \mathcal{B}.$$

Korollar

Ist $\mathcal{A} \leq_m \mathcal{B}$ und $\mathcal{A}$ nicht entscheidbar, dann ist auch $\mathcal{B}$ nicht entscheidbar.

Definition: Halteproblem

$$HALT = \{\langle\mathcal{M}\rangle w \in \Sigma^* | \mathcal{M} \text{ hält bei Eingabe } w\}$$

Lemma

Das Halteproblem $HALT$ ist nicht entscheidbar.

Aufgabe

Zeigen Sie, dass die Sprache
$\mathcal{L} = \{\langle\mathcal{M}\rangle \mid \text{Turingmaschine } \mathcal{M} \text{ akzeptiert jede Eingabe}\}$
nicht entscheidbar ist!

Aufgabe

Geben Sie, sofern möglich, je eine Lösung für die folgenden Post-Systeme an! Begründen Sie gegebenenfalls, warum es keine Lösung geben kann!

1. $$\left\{\begin{pmatrix} aa \\ a \end{pmatrix}, \begin{pmatrix} b \\ aa \end{pmatrix}, \begin{pmatrix} a \\ aab \end{pmatrix}\right\}$$

2. $$\left\{\begin{pmatrix} 01 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 101 \end{pmatrix}, \begin{pmatrix} 101 \\ 0 \end{pmatrix}\right\}$$

Reflexion

Was haben wir heute gelernt?

Reflexion

Was haben wir heute gelernt?

- Übungsblatt 4 besprochen

Reflexion

Was haben wir heute gelernt?

- Übungsblatt 4 besprochen
- Wortproblem in der Chomskyhierarchie

Reflexion

Was haben wir heute gelernt?

- Übungsblatt 4 besprochen
- Wortproblem in der Chomskyhierarchie
- many-one Reduktion kennengelernt

Reflexion

Was haben wir heute gelernt?

- Übungsblatt 4 besprochen
- Wortproblem in der Chomskyhierarchie
- many-one Reduktion kennengelernt
- berechenbar, entscheidbar, nicht entscheidbar

Noch Fragen?

Vorschau

Vorschau

- Rekursionstheorem

Vorschau

- Rekursionstheorem
- ...

Bis zum nächsten Mal