Einführung in die Informatik IV

Aufgabensteller: Mark Scharbrodt
Abgabetermin: Do, 15. Juni, 12:00 Uhr

Tutoraufgabe 1

Was bewirkt der Einsatz von Speichern bei Automaten? Bekannt ist, dass durch Hinzunahme eines Kellers zu einem endlichen Automaten ein Kellerautomat entsteht. Auf diese Weise können genau die *kontextfreien* Sprachen erkannt werden. Was passiert, wenn nun ein zweiter Keller hinzugenommen wird?

a) Definieren Sie einen nichtdeterministischen Kellerautomaten mit zwei unabhängigen Kellern und die von solchen Automaten akzeptierte Sprache.

b) Zeigen Sie, dass jede Sprache, die von einer Turingmaschine erkannt wird, auch von einem Kellerautomaten mit zwei Kellern erkannt wird.

Hausaufgabe 1 (freiwillig)

Geben Sie eine Turingmaschine an, die für alle Eingaben aus $a^\star b^\star c^\star$ hält. Als Ergebnis liefert die Turingmaschine genau dann das Zeichen `y` auf dem Band zurück, wenn die Eingaben aus $\{a^m b^{m+1} c^{2m} \mid 0 \leq m\}$ war.
Anderenfalls soll das Zeichen `n` auf das Band geschrieben werden. Es kann davon ausgegangen werden, dass sich zu Beginn der Kopf der Lesemaschine unter dem ersten Zeichen der Eingabe befindet. Am Schluss soll der Lesekopf unter dem `y` bzw. dem `n` stehen.
Beschreiben Sie, wie eine bandbeschränkte Turingmaschine arbeiten müßte, falls anstelle von `y` der Wert von m in Dezimaldarstellung auf dem Band zurückgegeben werden soll. Erläutern sie, warum eine bandbeschränkte Turingmaschine ausreicht.

Hausaufgabe 2 (freiwillig)

Für Typ 1 Grammatiken war die Bedingung, dass die rechten Regelseiten nicht kürzer sind, als die linken. Für diese Grammatiken lässt sich eine Normalform angeben, die in gewisser Weise mit der Chomsky Normalform bei den kontextfreien Grammatiken vergleichbar ist.

Definition. Eine Typ 1 Grammatik ist in *Kuroda Normalform*, falls alle Regeln eine der vier Formen haben:

$$A \to a, A \to B, A \to BC, AB \to CD.$$

Dabei stehen A, B, C, D für je eine Variable und a für ein Terminalsymbol.

a) Geben Sie eine Typ 1 Grammatik G in Kuroda Normalform an, welche die kontextsensitive Sprache $L(G) = \{a^n b^n c^n \mid n \in \mathbb{N}\}$ erzeugt.

b) Zeigen Sie, dass für jede Typ 1 Grammatik G mit $\epsilon \notin L(G)$ eine Grammatik G' in Kuroda Normalform existiert mit $L(G) = L(G')$.

Hausaufgabe 3 (2 + 3 = 5 Punkte)

Abschlusseigenschaften von Sprachen.
Zeigen Sie:

a) Die kontextfreien Sprachen sind nicht abgeschlossen unter Komplementbildung.

b) Sei $L \subset \Sigma^*$. Dann ist

$$MIN(L) = \{w \in L \mid \text{ kein echtes Präfix von } w \text{ ist in } L\}.$$

Zeigen Sie, dass die Klasse der deterministisch kontextfreien Sprachen unter MIN abgeschlossen ist. (Hinweis: Betrachten Sie die analogen Kellerautomaten)

Hausaufgabe 4 "CS-Parse" (5 Punkte)

Der in dieser Aufgabe programmierte Algorithmus wird wieder mit dem SOLEA–Tool korrigiert, d.h. alle diesbezüglichen Konventionen sind einzuhalten. Der Algorithmus soll für gegebene ϵ–freie Typ 1 Grammatiken G testen, ob gegebene Worte w in $L(G)$ enthalten sind. Das Programm soll von der Standardeingabe die Grammatiken und die Wörter einlesen und entsprechend auf der Standardausgabe eine Folge von `yes` oder `no` ausgegeben, je nachdem, ob die Worte in der Grammatik enthalten sind, oder nicht.
Da es sich um allgemeine kontextsensitive Grammatiken handelt, kann der CYK–Algorithmus nicht verwendet werden. Vielmehr soll das Programm versuchen, die eingelesenen Worte durch vollständiges "Durchprobieren" der Produktionen der Grammatik zu erzeugen. Natürlich begrenzt die Wortlänge die Zahl der durchzumusternden Produktionen.

Das Eingabeformat entspricht dem Eingabeformat von Hausaufgabe 4 Blatt 3. Die erste Zeile enthält die Anzahl der zu bearbeitenden Grammatiken. Jede Grammatik hat das folgende Format: Die erste Zeile enthält alle Variablen, jede Variable als einzelnen Grossbuchstaben, wobei S immer die Startvariable bezeichnet. In der zweiten Zeile stehen die Terminalzeichen (jeweils ein Kleinbuchstabe). Es folgt die Anzahl der Produktionen als Integerwert und dann in jeder Zeile eine Produktion der Form Ab CeF ($\Leftrightarrow Ab \to CeF$). Die letzte Zeile enthält das zu überprüfende Wort.
Beispiel: Die Eingabe

```
2
SB
abc
5
S aBSc
S abc
Ba aB
Bc bc
Bb bb
aabcc
SPMX
apm
6
S PX
S a
S MX
X SS
P p
M m
ppaamaa
```

soll die Ausgabe

```
no
yes
```

produzieren.
Hinweise:

- Da die Laufzeit dieses Verfahrens exponentiell ist, sollten Sie für Ihre Testläufe wie im obigen Beispiel nur sehr kurze Worte (z. B. ≤ 10 Buchstaben) und auch nur kleine Grammatiken verwenden.

- Zur Verkürzung der Laufzeit ist es sinnvoll, Worte, die im Ableitungsbaum bereits erzeugt wurden, nicht mehrfach zu expandieren.

- Der Name des Programmes lautet "CS–PARSE". Dieser sollte zusammen mit Ihrer Matrikelnummer in der ersten Zeile Ihres Programmes stehen (siehe Hinweise im Web).

- Auf der WWW-Seite der Vorlesung ist eine Beschreibung zu finden, wie die Lösung per Email eingereicht werden kann. Zudem finden Sie dort auch ein Programmgerüst, das schon wesentliche Teile des Codes (Datenstrukturen, Einleseroutinen etc.) enthält. Wenn Sie dieses Programmgerüst verwenden wollen, müssen dann nur noch die Routinen geschrieben werden, um die Grammatik zu durchmustern. Unser Tool zur automatischen Korrektur der Lösung wird dann natürlich nur diesen Teil mit anderen abgegebenen Lösungen vergleichen, nicht aber das bereits gegebene Programmgerüst.