



6. Übungsblatt Informatik II (Abgabe: 30.05.2005)

Organisatorische Hinweise:

- Die 1. Klausur findet am 18.06.05 von 10-13 Uhr statt.
 - Bitte denkt daran, daß Zulassungsvoraussetzung zu dieser Klausur das Erreichen von 50% der Übungspunkte der Übungen 1 bis 7 und das einmalige Vorrechnen in der Übungsgruppe bis zum 17.06.05 sind.
 - Klausurrelevant wird der Stoff der Vorlesung bis zum 15.06.05, sowie der Inhalt der Übungsblätter 1 bis 7 sein.
-

1. Aufgabe: (Ableitungsbaum) (4 + 4 + 4 Punkte)

Gib für folgende Programme den Ableitungsbaum an, sofern sie in $C0$ liegen, bzw begründe warum sie nicht in $C0$ liegen.

Hinweis: Schreibe im Querformat

- (a)

```
int a;
int b;
int main(int c)
{
int d;
if (a==b) then {b = 5}
else {b = 0; e = a};
return d}
a=main(3)
```
- (b)

```
int main(){ return 4};
int funktion(int x){return 9*x};
main(4)
```
- (c)

```
int a(){return 0};a=b()
```

2. Aufgabe: (Grammatiken für kontextfreie Sprachen) (3 + 3 Punkte)

Gib jeweils eine kontextfreie Grammatik für folgende Sprachen an.

- (a) $L_1 = \{a^i \mid i > 0\}$
(b) $L_2 = \{a^i b^j c^j \mid i, j > 0\}$

3. Aufgabe: (if then else) (5 Punkte)

Der Ableitungsbaum für

if BA then if BA then An else An

ist ohne Klammern nicht eindeutig. Schreibe ein Programm, mit dem Du überprüfen kannst, wie ein gegebener Compiler die obige Anweisung klammert. Überprüfe dann mit einem beliebigen C Compiler wie dieser die obige Anweisung klammert. Gib an, welchen Compiler Du verwendet hast.



6. Übungsblatt Informatik II (Abgabe: 30.05.2005)

4. Aufgabe: (Minuszeichen) (5 Punkte)

Um Entscheiden zu können, ob eine gegebene Zeichenreihe in einer gegebenen Grammatik liegt, muß ein Compiler unäre von binären Minuszeichen unterscheiden können. Gegeben: die einfache Grammatik G aus der Vorlesung mit den Ableitungen:

$$AA \rightarrow VC|AA + AA|AA -_2 AA|AA * AA|AA/AA| -_1 AA|(AA)$$

Terminale der Grammatik: $\{VC, (,), +, -_2, -, -_1, *, / \}$

Nichtterminale: $\{AA\}$

Zeige durch Induktion über die Ableitungsbäume:

- links von $-_2$ steht VC oder $)$
- links von $-_1$ steht $+, -_2, *, / -_1, (, '$ das leere Wort'

Hinweis: Zeige zuerst mit welchen Zeichen ein Wort aus der durch die Grammatik definierten Sprache enden kann.

5. Aufgabe: (uvwxy Theorem) (7* Punkte)

Das Haupthilfsmittel um zu zeigen, daß eine Sprache nicht kontextfrei ist, ist das folgende Theorem:

Satz: (uvwxy Theorem, Pumping Lemma für kontextfreie Sprachen)

Sei G die Grammatik einer kontextfreien Sprache. Dann gibt es eine Zahl $n \in \mathbb{N}$, so daß sich alle Wörter $z \in L(G)$ mit $|z| \geq n$ in $z = uvwxy$ zerlegen lassen und folgendes gilt:

- $|vx| \geq 1$
- $|vwx| \leq n$
- $\forall i \geq 0 : uv^iwx^iy \in L(G)$

Benutze das uvwxy Theorem um zu beweisen, daß die folgende Familie von Programmen, bei denen jede benutzte Variable vorher deklariert wurde, nicht kontextfrei ist. Hinweis: $a^4 = aaaa$

$$P(n): \text{int } a^n b^n; a^n b^n = 1$$

Um dies zu zeigen, wenden wir das uvwxy Theorem auf das Wort $\text{int } a^N b^N; a^N b^N = 1$ an und zeigen, daß sich das Wort nicht gemäß dem Theorem zerlegen läßt und somit die Sprache nicht kontextfrei ist. Hierbei soll n das n aus dem Theorem sein und es soll gelten $N > n$. Vorgehensweise: Zerlege das Wort so, daß a.) und b.) erfüllt sind und zeige, daß c.) nicht gilt. Hinweis: Fallunterscheidung:

ein Fall : $vwx = a^j$ mit $j < n$: uv^iwx^iy liegt nicht in der Sprache, da die Anzahl der a vor dem Strichpunkt größer der Anzahl der a nach dem Strichpunkt ist.

Zeige die restlichen Fälle.