

36 Punkte Aufgabe 1 (Lex und Yacc)

In dieser Aufgabe sollen Zahlwörter in Dezimalwerte übersetzt werden. Bis zur Einführung des Dezimalsystems im Jahre 1971 bestand ein englisches Pfund aus 20 Shilling oder 240 Penny. Die Währung bestand also aus der Stückelung Pfund, Shilling und Penny. Ziel ist es nun, ein Programm zu erstellen, welches eine Folge von englischen Geldbeträgen mittels Lex und Yacc übersetzt. Nachfolgend ein Beispiel für eine Eingabe und deren Übersetzung:

five hundred twenty-one Pound six Shilling eleven Pence => 521 P, 6 S, 11 p

Folgende Syntax-Regeln sind dabei für den Eingabedatenstrom einzuhalten:

- (R1) Zur Darstellung von Geldbeträgen sind nur die Wörter
 one, two, ... nine,
 ten, eleven .. nineteen,
 twenty, thirty, ... ninety,
 sowie die Multiplikatoren „hundred“ und „thousand“ erlaubt. Zusätzlich können die Stückelungen „Pound“ oder „Pounds“, „Shilling“ und „Penny“ oder „Pennies“ vorkommen.
- (R2) Aufeinander folgende Geldbeträge werden durch das Zeichen * (asterisk) getrennt, auch mehrere zusammenhängende Trennsymbole sind erlaubt.
- (R3) Teilworte im Bereich 21-99 müssen durch das Zeichen - (dash), wie bei „twenty-one“, verbunden sein.
- (R4) Der Betrag vor dem Faktor „hundred“ darf Werte im Bereich 1-9 und vor dem Faktor „thousand“ im Bereich 1-999 annehmen. Penny-Werte sollten nur im Bereich 0-11, Shilling-Werte nur im Bereich 0-19 liegen.
- (R5) Im Eingabestrom darf zwischen „Hundertern“ und „Zehnern“ auch optional das Wort „and“ stehen. Zum obigen Beispiel äquivalent wäre also „fivehundred **and** twenty-one ...“.

7 Punkte

- (a) Stellen Sie, unabhängig von der Aufgabenstellung, die grundsätzliche Arbeitsweise von Lex und Yacc dar. Was ist zu tun, um ein ablauffähiges Programm zu erhalten? Erläutern Sie die Zusammenhänge.

11 Punkte

- (b) Schreiben Sie ein Input-File für Lex, es soll folgende Token zurückgeben: UZEHN, UZWANZIG, UHUNDERT, HUNDERTMAL, TAUSENDMAL, POUND, SHILLING, PENNY, DASH, AND, ASTERISK. Zeilenumbrüche, Tabulatorzeichen und Leerzeichen im Eingabestrom sollen „verschluckt“

werden. Token, die mit U... beginnen, sollen der Variablen `yylval` einen Wert zuweisen, damit diese von Yacc ausgewertet werden kann.

- (c) Schreiben Sie ein Input-File für Yacc, welches den Eingabestrom syntaktisch analysiert. Überlegen Sie sich dazu zunächst geeignete Produktionsregeln für folgende Nichtterminalsymbole: `zahlen`, `zahl`, `umillion`, `utausend`, `uhundert`, `uzwoelf`, `uzwanzig`. Bei Verletzungen der Wertebereiche von Penny und Shilling soll eine Warnung ausgegeben werden. Für Ausgaben ist die C-Funktion `printf(const char *format, ...)` zu verwenden. 18 Punkte

Aufgabe 2 (Reguläre Ausdrücke, Grammatiken)

12 Punkte

Gegeben sei die Grammatik $G = (N, \Sigma, P, S)$ mit

$$N = \{S, A, B, C, D, E, F\},$$

$$\Sigma = \{a, b, c\} \text{ und}$$

$$P = \{ S \rightarrow AE \mid D,$$

$$E \rightarrow BF \mid D,$$

$$F \rightarrow AF \mid \epsilon,$$

$$A \rightarrow a,$$

$$B \rightarrow b,$$

$$C \rightarrow c \quad \}.$$

- (a) Geben Sie einen regulären Ausdruck α an, so daß $L(G) = L(\alpha)$ gilt. 2 Punkte
- (b) Geben Sie zu G eine reduzierte Grammatik G' an mit $L(G)=L(G')$ 5 Punkte
- (c) Geben Sie einen endlichen Automaten M an, der die Sprache $L(\alpha)$ akzeptiert. Die Übergangsfunktion δ soll dabei in Form eines Zustandsdiagramms dargestellt werden. 5 Punkte

36 Punkte Aufgabe 3 (FIRST- und FOLLOW-Mengen)

Gegeben sei die folgende LL(1)-Grammatik $G = (N, \Sigma, P, S)$ mit

$$N = \{S, A, B, C, D, E\}$$

$$\Sigma = \{\mathbf{u}, \mathbf{v}, \mathbf{w}\} \text{ und}$$

$$P = \{ S \rightarrow A \mid B \mid C,$$

$$A \rightarrow \mathbf{u}BD \mid \mathbf{u}BE,$$

$$B \rightarrow B\mathbf{w} \mid \mathbf{v},$$

$$C \rightarrow D\mathbf{v} \mid \mathbf{w},$$

$$D \rightarrow \mathbf{u} \mid \varepsilon,$$

$$E \rightarrow \mathbf{w}B\mathbf{u}E \mid \varepsilon \}$$

- 8 Punkte (a) Überführen Sie die angegebene Grammatik in eine äquivalente LL(1)-Grammatik G' .
- 14 Punkte (b) Berechnen Sie die initialen Steuermengen mit dem im Kurs angegebenen Algorithmus. Geben Sie dabei auch den Graphen, der die Berechnungsreihenfolge bei der Bestimmung der FIRST-Mengen festlegt, an.
- 14 Punkte (c) Bestimmen Sie die Steuermengen für die Produktionen aus G' . Berechnen Sie dazu die FOLLOW-Mengen mit dem Algorithmus aus dem Kurstext und geben Sie den Graphen an, der dabei erzeugt wird.

16 Punkte Aufgabe 4 (Datenflußgleichungen)

Gegeben sei das folgende Programmstück im 3-Adreß-Code:

```
(1) z := x + y
(2) if x < 2 then goto (8)
(3) if y > 10 then goto (7)
(4) x := x - 1
(5) y := y + 1
(6) goto (1)
(7) if y < z then goto (1)
(8) x := x + 2
(9) z := y
```

- 3 Punkte (a) Bestimmen Sie die Basisblöcke zu diesem Programmstück.
- 3 Punkte (b) Geben Sie den zugehörigen Flußgraphen an.
- 10 Punkte (c) Ermitteln Sie die Mengen $gen(B_i)$, $kill(B_i)$, $in(B_i)$ und $out(B_i)$ zur Bestimmung der UD-Verkettung.