Musterlösung

Aufgabe 1

a) Da die Signatur bewusst so vergeben wird, dass ein Exemplar eines Buches eindeutig identifiziert werden kann handelt es sich um einen (künstlichen) Schlüssel.

b) $\Sigma = \{"A"; "B"; "C"; \dots; "Z"; "0"; "1"; \dots; "9"; " "; "-"\}$

Anm: Umlaute sowie Unterscheidung von Groß- und Kleinschreibung sind zusätzlich denkbar, aber zur Lösung der Aufgabe nicht nötig.

c)

$$sig = bst, bst, bst, \{bst\}, " ", nr, ["-", nr]$$
$$bst = ("A" | "B" | "C" | \dots | "Z")$$
$$nr = ziffON, \{ziff\}$$
$$ziffON = ("1" | "2" | "3" | \dots | "9")$$
$$ziff = ("0" | ziffON)$$

(Zahlen ohne führende Nullen)

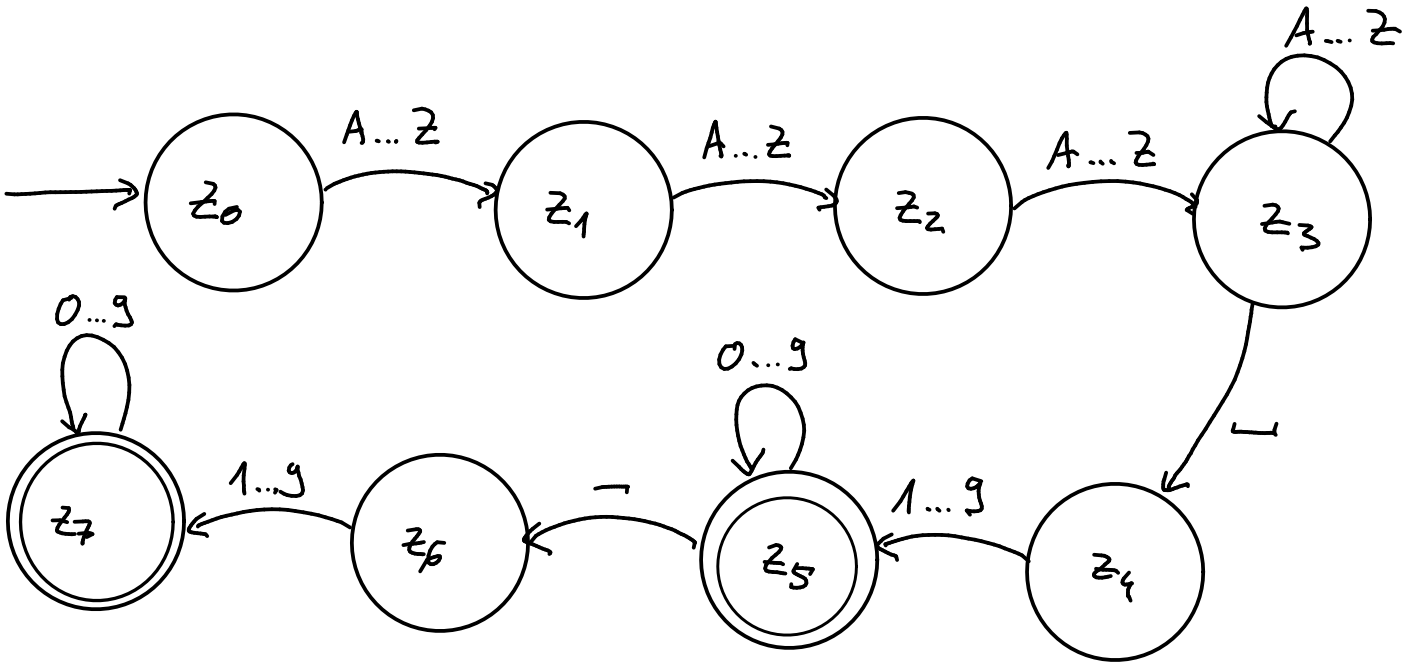
d) Eine Grammatik benötigt das Alphabet der Terminalsymbole, die Menge der Produktionsregeln sowie die Menge der Variablen (Nichtterminalsymbole) und das daraus hervorgehobene Start-symbol.

⇒ Menge der Variablen $V = \{sig; bst; nr; ziffON; ziff\}$

Startsymbol $S = sig$

Anm: selbstverständlich stark von Teilaufgabe c) abhängig

e)



mind 3 Bsf. ✓ (z0 – z3)

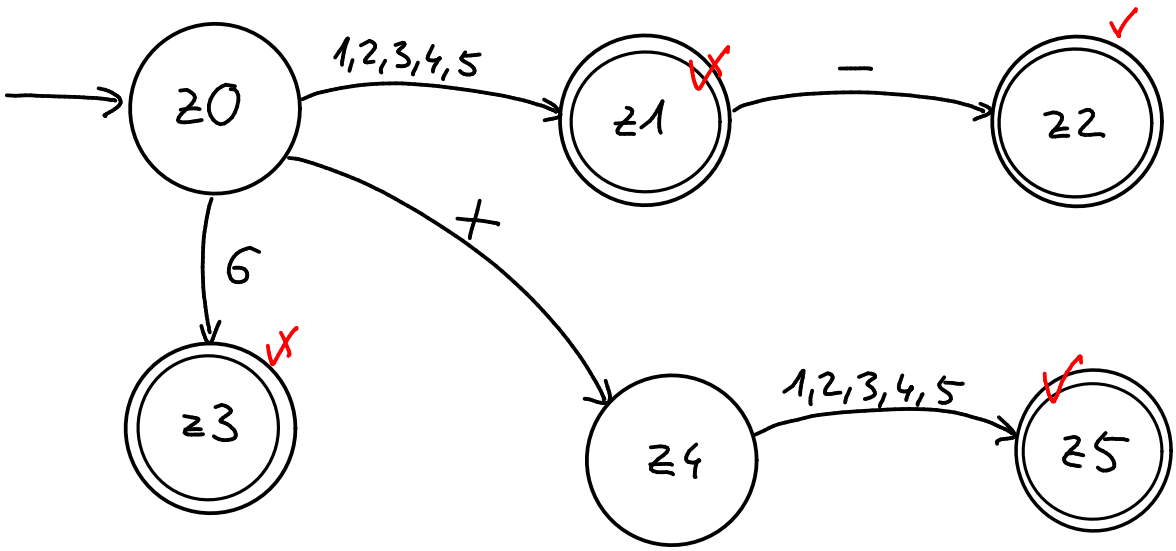
Zahl mit Endzust. ✓✓ (z4, z5)

Exemplarnummer: ✓ (z6, z7)

Aufgabe 2

a) Mit Syntax ✓ bezeichnet man die korrekte Schreibweise der Note, z.B. steht das „+" hier vor der Note. Die Semantik ✓ ist die Interpretation der Noten. Beispielsweise wird hier mit einer „+2" eine gute Leistung bewertet, die sogar noch zu einer sehr guten tendiert.

b)



c)
```
class Notenerkenner {
    String zustand;
    boolean testeNote(String eingabe) {
        zustand = "z0";
        for (int i=0; i< eingabe.length(); i++) {
            leseZeichen(eingabe.charAt(i));
        }
        if (zustand.equals("z1") || zustand.equals("z2") ||
            zustand.equals("z3") || zustand.equals(z5)) {
                return true;
            } else { return false;}
    }
    void leseZeichen(char c) {
        switch (zustand) {
            case "z0":
                switch (c) {
                    case '1': zustand = "z1"; break;
                    case '2': zustand = "z1"; break;
                    case '3': zustand = "z1"; break;
                    case '4': zustand = "z1"; break;
                    case '5': zustand = "z1"; break;
                    case '6': zustand = "z3"; break;
                    case '+': zustand = "z4"; break;
                    default : zustand = "falsch"; break;
                }
                // weitere Zustände ...
            }
        }
    }
```

d) Eine mögliche Lösung ist es, die eigentliche Note mit dem Datentyp int abzuspeichern und die Tendenz getrennt davon in einer eigenen Variable. Diese könnte auch Typ int sein und folgende Werte haben:

-1 entspricht einer negativen Tendenz (-)
0 entspricht einer Note ohne Tendenz
1 entspricht einer positiven Tendenz (+)

vv je nach Lösung; viele andere Lösungen denkbar

Aufgabe 3:

a) Es wird für jede Kombination von zwei Mitgliedern überprüft, ob diese am selben Tag Geburtstag haben.
Damit ist die Laufzeit quadratisch zur Anzahl der Mitglieder

$$T \sim n^2$$

(Begr. nicht nötig)

b) für alle i zwischen 0 und n-1:
falls (mitglied[i].geburtstag gleich mitglied[i+1].geburtstag)
wahr zurückgeben vv
endefalls.
endefüralle
falsch zurückgeben, v

c) Da die Laufzeit des Sortierens proportional zu $n \cdot \ln(n)$ ist und das finden im sortierten Feld linear zu n skaliert, ist die Gesamtlaufzeit etwa $n \cdot \ln(n) + n = n(\ln(n)+1)$
Bei großen n ist das in jedem Fall geringer als n^2.

zB $n=100 \Rightarrow n(\ln(n)+1) \approx 561 < n^2 = 10000$