

Vorname

Nachname

Matrikel-Nr.

Diese Klausur besteht aus 5 Aufgaben. Schreiben Sie jede Lösung auf die Vorderseite eines *neuen* Blattes (und lassen Sie die Rückseiten Ihrer Lösungsblätter *leer*).

**Aufgabe 1 (20 Punkte):** Geben Sie eine (kontextfreie) Grammatik an für die Menge aller Zahlen im 6-er-System, die (ohne Rest) durch 4 teilbar sind.

**Aufgabe 2 (10 Punkte):** Geben Sie eine (kontextfreie, Typ2-) Grammatik für *binäre Bruchzahlen ohne unnötige Nullen* an. Beispiele für solche Bruchzahlen:

101.01    0.1    1.0    0.0    110011.10101    10000.000000001

Gegenbeispiele:

.01            Keine Ziffer vor dem Punkt  
 101.          Keine Ziffer nach dem Punkt  
 0101.01      Unnötige 0 vorn  
 101.010      Unnötige 0 hinten

**Aufgabe 3 (10 Punkte):** Ein *Bezeichner* muss mit einem Buchstaben beginnen. Danach dürfen 0 oder mehr Buchstaben und/oder Ziffern folgen. Als Buchstaben sind nur die drei Kleinbuchstaben a, b, c erlaubt, als Ziffern nur 0, 1. Beispiele für solche Bezeichner:

a1   ba   ababa   c1001   abcc001   a10bcb001a0   a

Geben Sie eine Typ-3-Grammatik (eine reguläre Grammatik) an für die Menge all dieser Bezeichner.

**Aufgabe 4 (20 Punkte):** In einem Gentle-Programm sei folgender Typ definiert:

```
1 'type' INTLIST
2   leer
3   list(Element: INT, Rest: INTLIST)
```

Programmieren Sie zwei Prädikate namens `plusMinus` und `anzahlPaare` entsprechend den folgenden Spezifikationen:

```
4 -----
5 'action' plusMinus(Rein: INTLIST -> Raus: INTLIST)
6   -- Jedes Element aus Rein soll in Raus zweimal vorkommen,
7   -- einmal mit dem gleichen Vorzeichen wie in Rein und einmal
8   -- mit dem umgekehrten Vorzeichen wie in in Rein. Ausnahme:
9   -- Wenn in Rein eine 0 vorkommt, soll in Raus dafuer auch nur eine
10  -- 0 vorkommen (nicht zwei).
11  -- Ein Beispiel (in einer Phantasie-Syntax notiert):
12  -- Rein: [-3, 5, 0, 3, 5, 0]
13  -- Raus: [-3, 3, 5, -5, 0, 3, -3, 5, -5, 0]
14 -----
15 'action' anzahlPaare(Rein: INTLIST -> AnzahlPaare: INT)
16  -- Liefert die Anzahl der Paare, die sich in Rein befinden.
17  -- Ein Paar besteht aus zwei benachbarten und gleichen Elementen.
18  -- Beispiele (in einer Phantasie-Syntax notiert):
19  -- Die INTLIST [2, 3, 2, 3, 5] enthaelt 0 Paare
20  -- Die INTLIST [2, 3, 3, 5, 2] enthaelt 1 Paar
21  -- Die INTLIST [2, 2, 5, 3, 3] enthaelt 2 Paare
22  -- Die INTLIST [3, 2, 2, 2, 3] enthaelt 1 Paar (nicht 2!)
23 -----
```

**Aufgabe 5 (20 Punkte):** Betrachten Sie die folgenden Gentle-Typen, Muster und Werte:

```

1  'type' TX x1 x2 x3
2  'type' TY y1 y2 y3
3  'type' LISTXY
4      e                -- e wie empty (leer)
5      k(TX, TY, LISTXY) -- k wie Knoten (einer Liste)
6
7  Muster M1: k(x3, Y, L)
8  Muster M2: k(x1, Y, e)
9  Muster M3: k(X, Y, L)
10 Muster M4: k(x1, y1, e)
11 Muster M5: k(X, y2, L)
12
13 Wert   W1: k(x2, y3, k(x3, y2, e))
14 Wert   W2: k(x3, y2, k(x2, y3, e))
15 Wert   W3: k(x1, y1, e)

```

Die Variable **X** ist vom Typ **TX**, **Y** ist vom Typ **TY** und **L** vom Typ **LISTXY**.

Welche Muster passen auf welche Werte? Tragen Sie alle *gelingenden* Abgleiche zwischen den Mustern **M1** bis **M5** und den Werten **W1** bis **W3** in die folgende Tabelle ein. Der schon vorhandene Eintrag gibt an, dass ein Abgleich zwischen dem Muster **M3** und dem Wert **W1** gelingt und welche Variablenbelegung sich als Ergebnis dieses Musterabgleichs ergibt.

Das Muster	Paßt auf den Wert	Dabei ergibt sich die Variablenbelegung:
M1	W2	{ (Y, y2), (L, k(x2, y3, e)) }

Möglicherweise brauchen Sie nicht alle Zeilen der Tabelle.

**Lösung 1 (20 Punkte):** Geben Sie eine (kontextfreie) Grammatik an für die Menge aller Zahlen im 6-er-System, die (ohne Rest) durch 4 teilbar sind.

R01: RK0 -> "0"  
 R02: RK1 -> "1"  
 R03: RK2 -> "2"  
 R04: RK3 -> "3"  
 R05: RK0 -> "4"  
 R06: RK1 -> "5"

R07: RK0 -> RK0 "0"  
 R08: RK1 -> RK0 "1"  
 R09: RK2 -> RK0 "2"  
 R10: RK3 -> RK0 "3"  
 R11: RK0 -> RK0 "4"  
 R12: RK1 -> RK0 "5"

R13: RK2 -> RK1 "0"  
 R14: RK3 -> RK1 "1"  
 R15: RK0 -> RK1 "2"  
 R16: RK1 -> RK1 "3"  
 R17: RK2 -> RK1 "4"  
 R18: RK3 -> RK1 "5"

R19: RK0 -> RK2 "0"  
 R20: RK1 -> RK2 "1"  
 R21: RK2 -> RK2 "2"  
 R22: RK3 -> RK2 "3"  
 R23: RK0 -> RK2 "4"  
 R24: RK1 -> RK2 "5"

R25: RK2 -> RK3 "0"  
 R26: RK3 -> RK3 "1"  
 R27: RK0 -> RK3 "2"  
 R28: RK1 -> RK3 "3"  
 R29: RK2 -> RK3 "4"  
 R30: RK3 -> RK3 "5"

**Lösung 2 (10 Punkte):** Geben Sie eine (kontextfreie) Grammatik für *binäre Bruchzahlen ohne unnötige Nullen* an.

R01: ZF -> "0"  
 R02: ZF -> "1"  
 R03: ZF -> ZF "0"  
 R04: ZF -> ZF "1"  
 R05: VOR -> "0"  
 R06: VOR -> "1" ZF  
 R07: NACH -> "0"  
 R08: NACH -> ZF "1"  
 R09: BRUCH -> VOR "." NACH

**Lösung 3 (10 Punkte):** Ein Bezeichner muss mit einem Buchstaben beginnen. Danach dürfen 0 oder mehr Buchstaben und/oder Ziffern folgen. Als Buchstaben sind nur die drei Kleinbuchstaben a, b, c erlaubt, als Ziffern nur 1, 2. Geben Sie eine *Typ-3-Grammatik* (eine *reguläre Grammatik*) an für die Menge all dieser Bezeichner.

Eine etwas umständliche Lösung mit *zwei* Zwischensymbolen:

R01: BEZ -> "a" -- abschliessend  
 R02: BEZ -> "b" -- abschliessend  
 R03: BEZ -> "c" -- abschliessend  
 R04: BEZ -> "a" REST -- rechtslinear  
 R05: BEZ -> "b" REST -- rechtslinear  
 R06: BEZ -> "c" REST -- rechtslinear

```

R07: REST -> "a"          -- abschliessend
R08: REST -> "b"          -- abschliessend
R09: REST -> "c"          -- abschliessend
R10: REST -> "0"          -- abschliessend
R11: REST -> "1"          -- abschliessend

R12: REST -> "a" REST     -- rechtslinear
R13: REST -> "b" REST     -- rechtslinear
R14: REST -> "c" REST     -- rechtslinear
R15: REST -> "0" REST     -- rechtslinear
R16: REST -> "1" REST     -- rechtslinear

```

Eine kürzere Lösungen mit nur *einem* Zwischensymbol:

```

R01: BEZ -> "a"          -- abschliessend
R02: BEZ -> "b"          -- abschliessend
R03: BEZ -> "c"          -- abschliessend

R04: BEZ -> "a" BEZ       -- rechtslinear
R05: BEZ -> "b" BEZ       -- rechtslinear
R06: BEZ -> "c" BEZ       -- rechtslinear
R07: BEZ -> "0" BEZ       -- rechtslinear
R08: BEZ -> "1" BEZ       -- rechtslinear

```

**Lösung 4 (20 Punkte): Die Prädikate plusMinus und anzahlPaare:**

```

16 -----
17 'action' plusMinus(Rein: INTLIST -> Raus: INTLIST)
18   -- Jedes Element aus Rein soll in Raus zweimal vorkommen,
19   -- einmal mit dem gleichen Vorzeichen wie in Rein und einmal
20   -- mit dem umgekehrten Vorzeichen wie in in Rein. Ausnahme:
21   -- Wenn in Rein eine 0 vorkommt, soll in Raus dafuer auch nur eine
22   -- 0 vorkommen (nicht zwei).
23   -- Ein Beispiel (in einer Phantasie-Syntax notiert):
24   -- Rein: [-3, 5, 0, 3, 5, 0]
25   -- Raus: [-3, 3, 5, -5, 0, 3, -3, 5, -5, 0]
26
27 'rule' plusMinus(leer -> leer): .
28 'rule' plusMinus(list(E, R1) -> list(E, R2)):
29   eq(E, 0)
30   plusMinus(R1 -> R2)
31 'rule' plusMinus(list(E, R1) -> list(E, list(-E, R2))):
32   plusMinus(R1 -> R2)
33 -----
34 'action' anzahlPaare(Rein: INTLIST -> AnzahlPaare: INT)
35   -- Liefert die Anzahl der Paare, die sich in Rein befinden.
36   -- Ein Paar besteht aus zwei benachbarten und gleichen Elementen.
37   -- Beispiele (in einer Phantasie-Syntax notiert):
38   -- Die INTLIST [2, 3, 2, 3, 5] enthaelt kein Paar
39   -- Die INTLIST [2, 3, 3, 5, 2] enthaelt ein Paar
40   -- Die INTLIST [2, 2, 5, 3, 3] enthaelt zwei Paare
41   -- Die INTLIST [3, 2, 2, 2, 3] enthaelt ein Paar (nicht zwei!)
42
43 'rule' anzahlPaare(leer -> 0): .
44 'rule' anzahlPaare(list(E, leer) -> 0): .
45 'rule' anzahlPaare(list(E1, list(E2, R)) -> AR+1):
46   eq(E1, E2)
47   anzahlPaare(R -> AR)
48 'rule' anzahlPaare(list(E1, list(E2, R)) -> AR):
49   anzahlPaare(list(E2,R) -> AR)
50 -- Alternative (und einfachere) Version der letzten Regel:
51 -- 'rule' anzahlPaare(list(E, R) -> AR):
52 --   anzahlPaare(R -> AR)
53 -----

```

**Lösung 5 (20 Punkte):** Die ausgefüllte Tabelle mit den Musterabgleichen:

Das Muster	Paßt auf den Wert	Daraus ergibt sich die Variablenbelegung:
M1	W2	$\{ (Y, y2), (L, k(x2, y3, e)) \}$
M2	W3	$\{ (Y, y1) \}$
M3	W1	$\{ (X, x2), (Y, y3), (L, k(x3, y2, e)) \}$
M3	W2	$\{ (X, x3), (Y, y2), (L, k(x2, y3, e)) \}$
M3	W3	$\{ (X, x1), (Y, y1), (L, e) \}$
M4	W3	$\{ \}$
M5	W2	$\{ (X, x3), (L, k(x2, y3, e)) \}$

Von den 5 X 3 gleich 15 Musterabgleichen gelingen die angegebenen 7.