

Vorname

Nachname

Matrikel-Nr.

Diese Klausur besteht aus **4 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 (20 Punkte): Geben Sie eine (kontextfreie, Typ-2-) Grammatik an für die deutschen Worte, mit denen man die Ganzzahlen von 20 bis 99 (einschließlich) üblicherweise bezeichnet. Startsymbol: ZW20_99 ("ZW" wie Zahl-Wort).

Beispiele für Worte dieser Sprache: "zwanzig", "einundzwanzig", "zweiundzwanzig", ..., "neunundzwanzig", "dreissig", "einunddreissig", ..., "vierzig", "einundvierzig", ..., "fuenfzig", ..., "sechzig", ..., "siebzig", ..., "achtzig", "neunzig", "einundneunzig", ... "neunundneunzig".

Anforderung: Ihre Grammatik darf nicht mehr als 20 Regeln umfasse und das Wort "und" darf nur *einmal* vorkommen!

Notation: Fassen Sie *Endsymbole* in doppelte Anführungszeichen ein (wie die Beispiele für Worte), um sie von den Zwischensymbolen zu unterscheiden. Benutzen Sie einen Doppelpunkt : als Trennsymbol (zwischen der linken und rechten Seite einer Regel, wie in Gentle).

Aufgabe 3 (30 Punkte): Betrachten Sie folgende Vereinbarung eines (Gentle-) Typs:

```
'type' LISTE2
  last(Elm: INT)
  list(Elm: INT, Rest: LISTE2)
```

Schreiben Sie ein Prädikat entsprechend der folgenden Spezifikation:

```
'action' Aus2mach1(L1: LISTE2, L2: LISTE2 -> Raus: LISTE2)
  -- Die Liste Raus enthaelt abwechselnd eine Zahl
  -- aus L1, dann eine aus L2, dann wieder eine aus L1
  -- und eine aus L2 etc. Falls L1 und L2 nicht gleich
  -- lang sind, werden "die ueberstehenden Zahlen der
  -- laengeren Liste" am Ende von Raus hintereinander
  -- eingefuegt(ohne Zahlen der anderen Liste dazwischen).
  -- Beispiel:
  -- Sei L1 gleich list(11, list(12, list(13, last(14))))
  -- und L2 gleich list(21, last(22))
  -- Dann ist Raus gleich
  -- list(11, list(21, list(12, list(22, list(13, last(14))))))
```

Aufgabe 4 (30 Punkte): In einem Gentle-Programm wurde der folgende Typ LISTE0 und das Prädikat Vert (wie "vertausche") vereinbart:

```
'type' LISTE0
  leer
  list(Element: INT, Rest: LISTE0)
```

```
'condition' Vert(Rein: LISTE0 -> Raus: LISTE0)
-- Raus ist eine Kopie von Rein, in der die ersten beiden
-- Elemente vertauscht wurden. Glückt nur, wenn Rein mindestens
-- 2 Elemente enthält.
'rule' Vert(list(E1, list(E2, R)) -> list(E2, list(E1, R)))
```

Angenommen, der Gentle-Ausführer hat die Variablen-Belegung VB0 (siehe Zeile Nr 1 der Tabelle) und führt damit den folgenden Prädikataufruf aus:

```
Vert(list(10, L23) -> list(ERST, REST))
```

Was passiert dann im Einzelnen? Füllen Sie die folgende Tabelle aus, wie in den letzten Vorlesungen besprochen und geübt:

Nr	Op	V/A/W/M	Variablenbelegung/Ausdruck/Wert/Muster
1	AW0	VB0	{ (L23, list(20, list(30, leer))) }
2		A0	
3	MA0	W0	
4		MU0	
5			
6			
7			
8			
9			

Aufgabe1:	Note:
Aufgabe1:	Datum:
Aufgabe1:	
Aufgabe1:	
Summe:	

Aufgabe1:	Note:
Aufgabe1:	Datum:
Aufgabe1:	
Aufgabe1:	
Summe:	

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 (20 Punkte): Geben Sie eine (kontextfreie, Typ-2-) Grammatik an für die deutschen Worte, mit denen man die Ganzzahlen von 20 bis 99 (einschließlich) üblicherweise bezeichnet. Startsymbol: **ZW20_99**.

Beispiele für Worte dieser Sprache: "zwanzig", "einundzwanzig", "zweiundzwanzig", ..., "neunundzwanzig", "dreissig", "einunddreissig", ..., "vierzig", "einundvierzig", ..., "fuenfzig", ..., "sechzig", ..., "siebzig", ..., "achtzig", "neunzig", "einundneunzig", ... "neunundneunzig".

Anforderung: Ihre Grammatik darf nicht mehr als 20 Regeln umfasse und das Wort "und" darf nur *einmal* vorkommen!

Notation: Fassen Sie *Endsymbole* in doppelte Anführungszeichen ein (wie die Beispiele für Worte), um sie von den Zwischensymbolen zu unterscheiden. Benutzen Sie einen Doppelpunkt : als Trennsymbol (zwischen der linken und rechten Seite einer Regel).

```

R01: ZW20_90 : "zwanzig"
R02: ZW20_90 : "dreissig"
R03: ZW20_90 : "vierzig"
R04: ZW20_90 : "fuenfzig"
R05: ZW20_90 : "sechzig"
R06: ZW20_90 : "siebzig"
R07: ZW20_90 : "achtzig"
R08: ZW20_90 : "neunzig"

R09: ZW1_9 : "ein"
R10: ZW1_9 : "zwei"
R11: ZW1_9 : "drei"
R12: ZW1_9 : "vier"
R13: ZW1_9 : "fünf"
R14: ZW1_9 : "sechs"
R15: ZW1_9 : "sieben"
R16: ZW1_9 : "acht"
R17: ZW1_9 : "neun"

R18: ZW20_99 : ZW20_90
R19: ZW20_99 : ZW1_0 "und" ZW20_90

```

Lösung 3 (30 Punkte): Betrachten Sie folgende Vereinbarung eines (Gentle-) Typs:

```
'type' LISTE2
  last(Elem: INT)
  list(Elem: INT, Rest: LISTE2)
```

Schreiben Sie ein Prädikat entsprechend der folgenden Spezifikation:

```
'action' Aus2mach1(L1: LISTE2, L2: LISTE2 -> Raus: LISTE2)
  -- Die Liste Raus enthaelt abwechselnd eine Zahl
  -- aus L1, dann eine aus L2, dann wieder eine aus L1
  -- und eine aus L2 etc. Falls L1 und L2 nicht gleich
  -- lang sind, werden "die ueberstehenden Zahlen der
  -- laengeren Liste" am Ende von Raus hintereinander
  -- eingefuegt(ohne Zahlen der anderen Liste dazwischen).
  -- Beispiel:
  -- Sei L1 gleich list(11, list(12, list(13, last(14))))
  -- und L2 gleich list(21, last(22))
  -- Dann ist Raus gleich
  -- list(11, list(21, list(12, list(22, list(13, last(14))))))

'rule' Aus2mach1(last(E1)      , L2      -> list(E1, L2))      : .
'rule' Aus2mach1(list(E1, R1), last(E2)  -> list(E1, list(E2, R1 ))) : .
'rule' Aus2mach1(list(E1, R1), list(E2, R2) -> list(E1, list(E2, R12))):
  Aus2mach1(R1, R2 -> R12)
```

Lösung 4 (30 Punkte): In einem Gentle-Programm wurde der folgende Typ `LISTE0` und das Prädikat `Vert` (wie "vertausche") vereinbart:

```
'type' LISTE0
  leer
  list(Element: INT, Rest: LISTE0)
'condition' Vert(Rein: LISTE0-> Raus: LISTE0)
  -- Raus ist eine Kopie von Rein, in der die ersten beiden
  -- Elemente vertauscht wurden. Glückt nur, wenn Rein mindestens
  -- 2 Elemente enthält.
  'rule' Vert(list(E1, list(E2, R)) -> list(E2, list(E1, R)))
```

Angenommen, der Gentle-Ausführer hat die Variablen-Belegung `VB0` (siehe Zeile Nr 1 der Tabelle) und führt damit den folgenden Prädikataufruf aus:

```
Vert(list(10, L23) -> list(ERST, REST))
```

Was passiert dann im Einzelnen? Füllen Sie die folgende Tabelle aus, wie in den letzten Vorlesungen besprochen und geübt:

Nr	Op	V/A/W/M	Variablenbelegung/Ausdruck/Wert/Muster
1	AW0	VB0	{ (L23, list(20, list(30, leer))) }
2		A0	list(10, L23)
3	MA0	W0	list(10, list(20, list(30, leer)))
4		MU0	list(E1, list(E2, R))
5	AW1	VB1	{ (L23, list(20, list(30, leer))), (E1, 10), (E2, 20), (R, list(30, leer)) }
6		A1	list(E2, list(E1, R))
7	MA1	W1	list(20, list(10, list(30, leer)))
8		MU1	list(ERST, REST)
9		VB2	{ (L23, list(20, list(30, leer))), (E1, 10), (E2, 20), (R, list(30, leer)), (ERST, 20), (REST, list(10, list(30, leer))) }