1. (4 Punkte) (a) Erkläre den Unterschied zwischen Syntax und Semantik einer Sprache.

   (b) Welche der folgenden Python-Zeilen enthalten einen Syntaxfehler? Was passiert bei den anderen beim Ausführen?

```
x = 3 +
print(10 / 0)
y = 'a' + 2
if x > 2 print(x)
z = [1, 2, 3][5]
```

2. (5 Punkte) Gegeben ist die Grammatik mit $V = \{S, B\}$, $\Sigma = \{a, b, c\}$, Startsymbol S und den Regeln
   $S \to aSc \mid B$
   $B \to bB \mid b$

   (a) Gib eine Linksableitung für das Wort *aabcc* an.

   (b) Zeichne den Ableitungsbaum zu *aabcc*.

   (c) Welche der Wörter *abc*, *aabc*, *abbbc*, *ac*, *aabbcc* gehören zur Sprache?

   (d) Beschreibe die Sprache in Mengenschreibweise.

3. (4 Punkte) Gegeben ist die Grammatik mit $V = \{S, A, B\}$, $\Sigma = \{a, b\}$, Startsymbol S und den Regeln

   $S \to AB$
   $A \to aS \mid a$
   $B \to bA$

   (a) Welche der Wörter *aba*, *abab*, *abaaba*, *aabba*, *aaababababa* gehören zur Sprache? Gib für ein Wort, das dazugehört und mehr als drei Zeichen hat, eine Ableitung an.

   (b) Begründe, warum in jedem Wort der Sprache doppelt so viele a wie b vorkommen.

4. (3 Punkte) Gegeben sind das Alphabet $\Sigma = \{a, b, c\}$ und die Sprache $L = \{a^i b^j c^k \mid i, j, k > 0\}$. Gib eine Grammatik an, die L erzeugt.

5. (3 Punkte) Gegeben sind das Alphabet $\Sigma = \{a, b, c\}$ und die Sprache $L = \{a^i b^j c^{i+j} \mid i, j \geq 0,\ i + j > 0\}$. Gib eine Grammatik an, die L erzeugt.

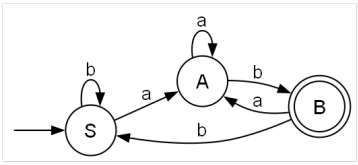
6. (3 Punkte) Gegeben ist das Alphabet $\Sigma = \{0, 1, 2\}$. Gib eine kontextfreie Grammatik für die Sprache $L = \{0^n 1^{2n} 2^m \mid n, m \geq 0\}$ an.

7. (4 Punkte) Gegeben ist das Alphabet $\Sigma = \{a, b, c, d\}$. L ist die Sprache aller Wörter, in denen nie ein b direkt auf ein a folgt, nie ein c direkt auf ein b und nie ein d direkt auf ein c. Zum Beispiel liegt *acadbbda* in L, *acab* nicht. Gib eine Grammatik an, die L erzeugt.

8. (3 Punkte) Beschreibe die Sprache, die von der folgenden Grammatik erzeugt wird: $V = \{S, A, B\}$, $\Sigma = \{0, 1\}$, Startsymbol S, Regeln:

   $S \to 0B \mid 1A$
   $A \to 0 \mid 0S \mid 1AA$
   $B \to 1 \mid 1S \mid 0BB$

9. (4 Punkte) Der DEA akzeptiert die Wörter über $\{a, b\}$, die auf ab enden.



   (a) Gib eine reguläre Grammatik an, die dieselbe Sprache erzeugt.

   (b) Gib eine Ableitung für *baab* an und erkläre, wie sie mit dem Lauf des Automaten zusammenhängt.

10. (3 Punkte) Gegeben ist die reguläre Grammatik mit $V = \{S, A\}$, $\Sigma = \{0, 1\}$, Startsymbol S und den Regeln

    $S \to 0S \mid 1A$
    $A \to 0A \mid 1S \mid \varepsilon$

    Zeichne einen endlichen Automaten, der dieselbe Sprache akzeptiert, und beschreibe die Sprache.

11. (5 Punkte) Gib für jede Grammatik den größten Typ in der Chomsky-Hierarchie an (3, 2, 1 oder 0) und begründe. Startsymbol ist jeweils S.

    G_1: $S \to aS \mid bA, \quad A \to b \mid \varepsilon$
    G_2: $S \to aSb \mid ab$
    G_3: $S \to aSBc \mid abc, \quad cB \to Bc, \quad bB \to bb$
    G_4: $S \to AB, \quad AB \to a$

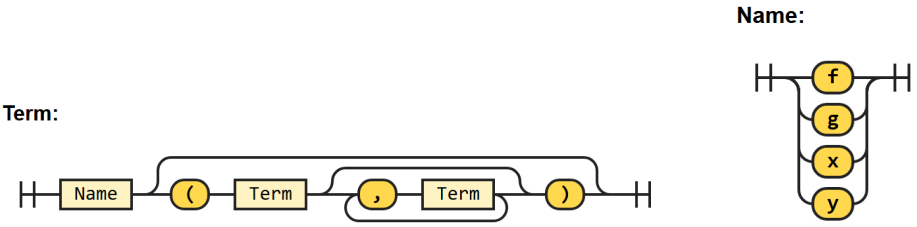
    (a) Typen von G_1 bis G_4

    (b) Welche Sprache erzeugt G_4? Welchen Typ hat diese Sprache?

12. (5 Punkte) Gegeben ist die Grammatik mit Startsymbol E und den Regeln $E \to E - E \mid Z$, $Z \to 1 \mid 2 \mid 3$.
    (a) Zeichne zwei verschiedene Ableitungsbäume für das Wort $3 - 2 - 1$.
    (b) Welche Werte ergeben sich, wenn man die Bäume als Rechenweg liest? Welcher Baum passt zur üblichen Rechnung?
    (c) Gib eine eindeutige Grammatik für dieselbe Sprache an, deren Ableitungsbaum zur üblichen Rechnung passt.

13. (4 Punkte) Gesucht ist eine Beschreibung für Python-Listen aus einzelnen Ziffern ohne Leerzeichen, z.B. `[]`, `[3]`, `[1,2,7]`. Nicht erlaubt sind z.B. `[1,]`, `[,]` und `[12]`.
    (a) Gib eine EBNF an. Du darfst `Ziffer ::= [0-9]` verwenden.
    (b) Zeichne das Syntaxdiagramm.

14. (5 Punkte) Gegeben sind die Syntaxdiagramme:



(a) Welche der folgenden Wörter sind ein `Term`? `f(x)`, `f()`, `g(x,f(y,x))`, `f(x,)`, `(x)`, `fg(x)`, `x`

(b) Gib eine EBNF an.

(c) Kann man die Sprache mit einem regulären Ausdruck beschreiben? Begründe.

15. (4 Punkte) Erweitere den Parser aus dem Notebook so, dass er auch Subtraktion und Division kann:

```
Ausdruck ::= Term ( ( '+' | '-' ) Term )*
Term     ::= Faktor ( ( '*' | '/' ) Faktor )*
```

Die Funktionen `faktor`, `zahl` und `auswerten` bleiben unverändert.

Was ergeben `auswerten('10-4-3')` und `auswerten('8/4/2')`?