Klausuraufgaben

1. a) S $\rightarrow$ Bb
   B $\rightarrow$ Bb (B erzeugt rechts überzählige b's)
   B $\rightarrow$ P
   S $\rightarrow$ aA
   A $\rightarrow$ aA (A erzeugt links überzählige a's)
   A $\rightarrow$ P
   P $\rightarrow$ aPb (P erzeugt Paare von a's und b's)
   P $\rightarrow$ ab

   b) Wäre L regulär, so auch das Komplement von L. Dieses ist aber die nicht reguläre Sprache $L = \{a^n b^n \mid n > 0\}$ (Beweis: L erfüllt Pumping Lemma nicht)

2. a) Teilmengenautomat
   b) Abschluss/ Nichtabschluss gegen Komplement

3. aaada wird von der Grammatik erzeugt

	a	a	a	d	a
a	A	-	-	-	S
a		A	-	S	B
a			A	B	D
d				D	-
a					A

aadaa wird nicht von der Grammatik erzeugt

	a	a	d	a	a
a	A	-	S	B	D
a		A	B	D	-
d			D	-	-
a				A	-
a					A

4. Darstellung in Tabellenform

		M			
		A	B	C	D
	A	x	ja	ja	ja
L	B	n	x	ja	ja
	C	n	?	x	ja
	D	nein	nein	nein	x

5.

1:	READ 1	(liest x1 in den Akku)
2:	STORE 1	(speichert x1 in Register 1)
3:	READ 0	(liest n in den Akku)
4:	READ $\rightarrow$ 0	(liest xn in den Akku)
5:	SUB 1	(bildet xn-x1 im Akku)
6:	JZERO 8	(Sind die Zahlen gleich?)
7:	HALT	(0 stand bereits im Akku)
8:	LOAD =1	(erzeugt 1 im Akku)
9:	HALT	

6. a) JA, denn: Λ und a definieren endliche Sprachen. Die Verkettung und die Vereinigung zweier endlicher Sprachen ist auch wieder endlich.
   b) NEIN. Λ^* oder $(a^*\Lambda)$ sind endliche Sprachen.

7. Sei A NP-vollständig und außerdem in P.

   Sei B eine beliebige Sprache in NP.

   Dann gilt B $\leq_{pol}$ A.

   Daraus folgt, dass auch B in P liegt.

   Also wäre NP eine Teilklasse von P.

   Umgekehrt ist P eine Teilklasse von NP.

8. Aus einem Algorithmus, der entscheidet, ob es in einem gegebenen Graphen einen Hamiltonpfad gibt, kann man wegen HC $\leq_{pol}$ HP einen Algorithmus machen, der entscheidet, ob es in einem gegebenen Graphen einen Hamiltonkreis gibt:

   Diesen verwenden wir nun, um der Reihe nach alle Kanten zu eliminieren, die für die Existenz eines Hamiltonkreises nicht erforderlich sind. Übrig bleiben nur Kanten, die in einem Hamiltonkreis auf jeden Fall vorkommen müssen. Diese bilden dann den gesuchten Hamiltonkreis.

   Die Laufzeit des Algorithmus, der einen Hamiltonkreis konstruiert, hat sich schlimmstenfalls um einen polynomiellen Faktor verschlechtert.

9. S $\rightarrow$ A

S → B

A → a

B → a

10. $S \rightarrow \epsilon \mid A \mid B \mid C \mid AB \mid AC \mid BC \mid ABC$

    $A \rightarrow a \mid aA \mid Aa \mid AaA$

    $A \rightarrow B \mid C \mid BC$

    $A \rightarrow aa$

    $B \rightarrow C \mid CC \mid CCC$

    $C \rightarrow c \mid Cc$

    $C \rightarrow dcd$

11. a) Ja, $S \rightarrow 0S1 \mid 0S0 \mid 1S1 \mid 1S0 \mid 1$
    b) Nein, L erfüllt das Pumping-lemma nicht
       0^n10^n in L
       Fall 1: v in einem der 0-Blöcke → $\mid x \mid \neq \mid y \mid$
       Fall 2: v=1 abpumpen → 0^{2n} nicht in L
       Fall 3: V = Kombination aus 0,1 → abpumpen 0^m nicht in L

12. a) Ja, $S \rightarrow 0S1 \mid 0S0 \mid 1S1 \mid 1S0 \mid 1 \mid 0$
    b) Ja, $\alpha = (00 \cup 01 \cup 11 \cup 10)^*(0 \cup 1)$

13. a) Nein, $S \rightarrow SS \rightarrow aSabSb \rightarrow aabb$, aabb ist kein Palindrom

b) Sq $\rightarrow$ aSaq
Sq $\rightarrow$ bSbq
Sq $\rightarrow$ SSq
sq $\rightarrow$ q
aqa $\rightarrow$ q
bqb $\rightarrow$ q

14. S $\rightarrow$ ϵ — S'

S' $\rightarrow$ XD — DS' — a — AB

A $\rightarrow$ DS' — a

B $\rightarrow$ b

D $\rightarrow$ a

X $\rightarrow$ DS'

15. nicht ableitbar

	a	a	a	d	a
a	A	-	-	-	-
a		A	-	-	-
a			A	S,B,C	-
d				D	-
a					A

ableitbar

	a	d	a	d	d
a	A	S,B,C	-	D,A	S,B,C
d		D	-	-	-
a			A	S,B,C	-
d				D	-
d					D

16. siehe Skript

17. a) Nein, beide Sprachen erzeugen nur Wörter mit einer geraden Anzahl a's, aber bei α_2 müssen die a's immer hintereinander stehen. Beispiel: aba in L_1, aber aba nicht in L_2

    b) siehe Skript

18. a) siehe Skript

    b)

r_0	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9
0	1	10	x_m	..	..	..	..	..	x_1

    c) es sortiert die Reihenfolge der eingegeben Register um

19. a) siehe Skript

    b) Wir wissen: es existiert eine Clique der größe k
    betrachte alle Knoten mit Grad $\geq k-1$

systematisches durchtesten eines Knotens mit allen seinen Nachbarn mit Grad $k-1$ oder gröer ob sie eine Clique ergeben

20. a) Man geht alle Möglichkeiten für cmax(x) durch bis das größte gefunden wurde.

    b) Nein, löst nicht das Maximierungsproblem.

21. a) siehe pic

    b)

	m	a
q0	q1	q0
q1	q1	q2
q2	q3	q0
q3	q1	q4
q3	q1	q4
q4	q3	q5
q5	q5	q5

22. a) siehe Script

    b) Ja, ...

23. Nein, sie erfüllt das Pummping-lemma nicht.

$$x * x - (x+1) * (x+1) = x * x - (x * x + 2x + 1) = -2x - 1$$

Zerlgung von a^n, $n = b^2$, in uvw, so dass uv^0w in L liegt: $v = a^(2b + 1)$
$\rightarrow uv^2w$ nicht in L

24. CS

    REG

    CF

    CF

25. a) siehe Script

    b) LL-Automat
       Sq → aSAq
       Aq → bAq
       Aq → aq
       bqb → q
       aqa → q

       LR-Automat
       ASaq → Sq
       Abq → Aq
       aq → Aq
       qa → aq
       qb → bq
       Sq → f

26. Endlicher Automat → nicht leistungsstrker
    Kellerautomat → leistungsstrker
    LBA → nicht bekannt
    Turingmaschine → nicht leistungsstrker

27. S → AX

    X → BD

    D → a

    A → a | b | BC

    B → b

    C → AB

28. nicht ableitbar

	a	a	b	c	d
a	A	-	-	-	-
a		A	S	-	-
b			B	A,S	C
c				C	-
d					D

ableitbar

	b	a	d	b	b
b	B	-	S,A	S	S
a		A	C	-	-
d			D	-	-
b				B	B
b					B

29. Kellerautomat hält nicht an
    Kellerautomat hat Wort zuende gelesen
    Keller ist leer
    Finalzustand ist erreicht

    Kellerautomat hält an
    Kellerautomat kann Wort nicht zuende lesen
    Keller ist leer
    Finalzustand ist erreicht

    Kellerautomat hält an
    Kellerautomat hat Wort zuende gelesen
    Keller ist nicht leer
    Finalzustand ist erreicht

    Kellerautomat hält an
    Kellerautomat hat Wort zuende gelesen
    Keller ist leer
    Finalzustand ist nicht erreicht

30. siehe Script

31. Die Sprache aller Wörter ber $\{a, b\}$.

32. ϵ-frei machen: S, A, B können alle zu ϵ abgeleitet werden, also müssen alle Möglichkeiten, in rechten Seiten einige der Nichtterminale zu löschen,

realisiert werden, und ein neues Startzeichen S' muss verwendet werden:

$S' \to S \mid \epsilon$
$S \to AaBS \mid aBS \mid AaS \mid AaB \mid Aa \mid aB \mid aS \mid a \mid BAA \mid AA \mid BA \mid B \mid A$
$A \to BB \mid B \mid aa$
$B \to BA \mid A \mid B$

Kettenproduktionen $S' \to S \to B \to A \to B$ einbauen

$S' \to \epsilon$
$S' \to AaBS \mid aBS \mid AaS \mid AaB \mid Aa \mid aB \mid aS \mid a \mid BAA \mid AA \mid BA$
$S \to AaBS \mid aBS \mid AaS \mid AaB \mid Aa \mid aB \mid aS \mid a \mid BAA \mid AA \mid BA$
$S' \to BB \mid aaS \to BB \mid aaA \to BB \mid aaB \to BB \mid aa$
$S' \to BAS \to BAB \to BAA \to BA$

Separieren

$S' \to \epsilon N \to a$
$S' \to ANBS \mid NBS \mid ANS \mid ANB \mid AN \mid NB \mid NS \mid a \mid BAA \mid AA \mid BA$
$S \to ANBS \mid NBS \mid ANS \mid ANB \mid AN \mid NB \mid NS \mid a \mid BAA \mid AA \mid BA$
$S' \to BB \mid NNS \to BB \mid NNA \to BB \mid NNB \to BB \mid NN$
$S' \to BAS \to BAB \to BAA \to BA$

Chomsky-NF

$S' \to \epsilon N \to aX \to ANY \to BSZ \to BA$
$S' \to XY \mid NY \mid XS \mid XB \mid AN \mid NB \mid NS \mid a \mid ZA \mid AA \mid BA$
$S \to XY \mid NY \mid XS \mid XB \mid AN \mid NB \mid NS \mid a \mid ZA \mid AA \mid BA$
$S' \to BB \mid NNS \to BB \mid NNA \to BB \mid NNB \to BB \mid NN$
$S' \to BAS \to BAB \to BAA \to BA$

33. Nicht regulär, da Pumping-Lemma verletzt: Quadratzahlen haben aber einen immer größer werdenden Anstand zur nächsten.

34. Ist kontextfrei: S $\rightarrow$ aSbb | abb

35. Verhaltensgleichheit berechnen (n^3-Algorithmus aus der Vorlesung)

    Prüfen, ob die beiden Startzustände verhaltensgleich sind.

    Bemerkung: Minimieren und Testen auf Isomorphie (Gleichheit bis auf Umbenennung) hilft nicht, da kein polynomieller Isomorphietest fr Graphen bekannt ist.

36. Endlicher Automat, der Teilbarkeit durch 8 mit 8 Zuständen feststellt. Daraus regulären Ausdruck machen.

37. Es gibt zwei Knoten mit ungeradem Grad. Alle anderen Knoten haben geraden Grad. Also: Eulerkreis nein, Eulerpfad ja

38. Zu A äquivalente kontextfreie Grammatik G in Chomsky-NF herstellen und mit CYK-Algorithmus testen, ob w in L(G) liegt

    A einfach laufen zu lassen, hilft nicht: A ist nicht unbedingt deterministisch, und selbst wenn doch, kann A auf w unendlich lange laufen.

39. Konjunktive Normalform: NP-vollständig
    Disjunktive Normalform: in P

40. In NP, weil:
    1. in polynomieller Zeit auf korrektes Eingabeformat testbar
    2. in polynomieller Zeit testbar, ob eine Partition 3 identische Summen liefert
    3. Existenzquantor: gibt es eine Partition mit ...
    4. Polynomielle Ausgeglichenheit: Partition ist nicht länger als der Input

    NP-vollständig, da PARTITION wie folgt darauf reduzierbar ist:
    Falls $s = x_1 + ... + x_n$ gerade ist, kann $x_1, ..., xxx$ in zwei gleich groe Teilsummen zerlegt werden, genau dann wenn $x_1, ..., xxx, s/2$ in drei gleich groe Teilsummen zerlegt werden kann.