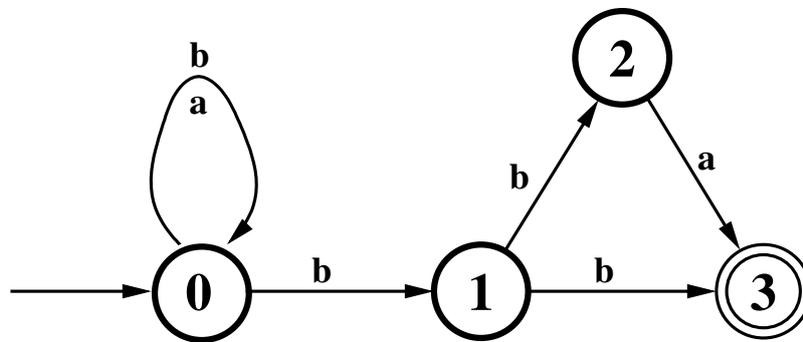


Aufgabe 1. (6 + 3 + 1 = 10 Punkte)

Konstruieren Sie zu dem folgenden Nichtdeterministischen Endlichen Automaten den äquivalenten Deterministischen Automaten. Wenden Sie dazu das aus der Vorlesung bekannte Verfahren an.

- (a) Verwenden Sie die Zustandsübergangstabelle zur Konstruktion des Deterministischen Endlichen Automaten.
- (b) Zeichnen Sie das Zustandsdiagramm des Deterministischen Endlichen Automaten (nächste Seite).
- (c) Geben Sie die Sprache, die von dem Automaten erkannt wird, als Regulären Ausdruck an.

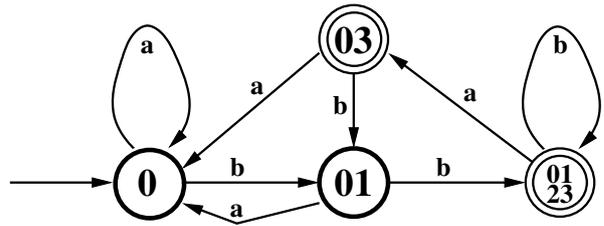


	a	b

Lösung a:

	a	b
{0}	{0}	{01}
{01}	{0}	{0123}
{0123}	{03}	{0123}
{03}	{0}	{01}

Lösung b:



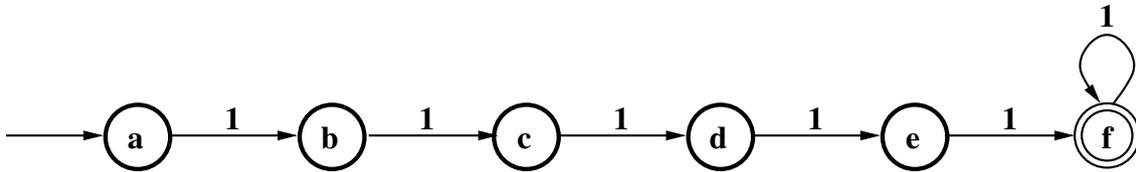
Der resultierende DEA hat also folgende Form:

Lösung c:

$$RA = (a + b)*b(b + ba)$$

Aufgabe 2. (5 Punkte)

Gegeben sei der folgende endliche Automat mit Eingabealphabet $\{1\}$ und Zustandsmenge $\{a, b, c, d, e, f\}$:



Geben Sie für jedes $k \in \{0, 1, 2, 3, 4\}$ jeweils die Mengen der Zustände an, die zueinander k -äquivalent sind.

Hinweis: Mengen mit nur einem Element müssen Sie hier nicht aufführen.

k	Mengen k -äquivalenter Zustände
0	
1	
2	
3	
4	

Lösung:

k=0: $\{a, b, c, d, e\}$

k=1: $\{a, b, c, d\}$

k=2: $\{a, b, c\}$

k=3: $\{a, b\}$

k=4:

Aufgabe 3. (8 Punkte)

Entwickeln Sie eine kontextfreie Grammatik, mit der sich alle Regulären Ausdrücke über der Variablenmenge $E = \{a, b, c\}$ und den Operatoren $O = \{+, \cdot, *\}$ mit korrekter Klammerung erzeugen lassen. Beim Punktoperator \cdot soll dabei auch die (den Operator weglassende) Kurzschreibweise möglich sein.

Hinweis: Die Grammatik muss vollständig angegeben werden, es reicht nicht, nur die Produktionen hinzuschreiben.

Lösung: $G = (N, T, P, S)$ mit

$$N = \{S\},$$

$$T = \{a, b, c, +, \cdot, *, (,), \emptyset\},$$

$$P = \left\{ \begin{array}{l} S \rightarrow (S + S) \mid (S \cdot S) \mid (SS) \mid S^*, \\ S \rightarrow a \mid b \mid c \mid \emptyset \end{array} \right\}.$$

Aufgabe 4. $((1 + 2 + 1 + 1 + 2) + 2 = 9$ Punkte)

(a) Geben Sie für die folgenden Komplexitätsklassen jeweils ein aus der Vorlesung bekanntes Problem an, das sicher in dieser Klasse liegt, oder begründen Sie, warum Sie keines angeben können.

(1) P

(2) $NP \setminus P$

(3) NP

(4) $\{L \mid L \text{ ist NP-vollständig}\}$

(5) $\{L \mid L \notin NP\}$

Lösung:

1. Wortproblem für Typ3-,
~Typ2-Sprachen:
 $\{(G, w) \mid G \text{ ist kontextfrei und } w \in L(G)\}$,
Sortieren,
Automatenminimierung, und andere.
2. keine Angabe möglich, Klasse leer, falls $P = NP$.
3. alle aus 1), alle aus 4).
4. SAT, 3-SAT, CLIQUE, und andere.

5. Halteproblem:

$\{(T, w) \mid \text{Turingmaschine } T \text{ hält auf Eingabe } w\}$,

Diagonalsprache,

$L_{NA} = \{w_i \in E^* \mid w_i \notin L(T_i)\}$, und andere.

(b) Warum werden Beweise der NP-Vollständigkeit im Prinzip im Lauf der Zeit immer einfacher?

Lösung:

- Die Anzahl der als NP-vollständig bekannten Probleme steigt.
- Es reicht, ein einziges NP-vollständiges Problem auf ein zu untersuchende Problem zu reduzieren, um NP-Vollständigkeit zu beweisen. Daher hat man mit der Zeit immer mehr Auswahl.

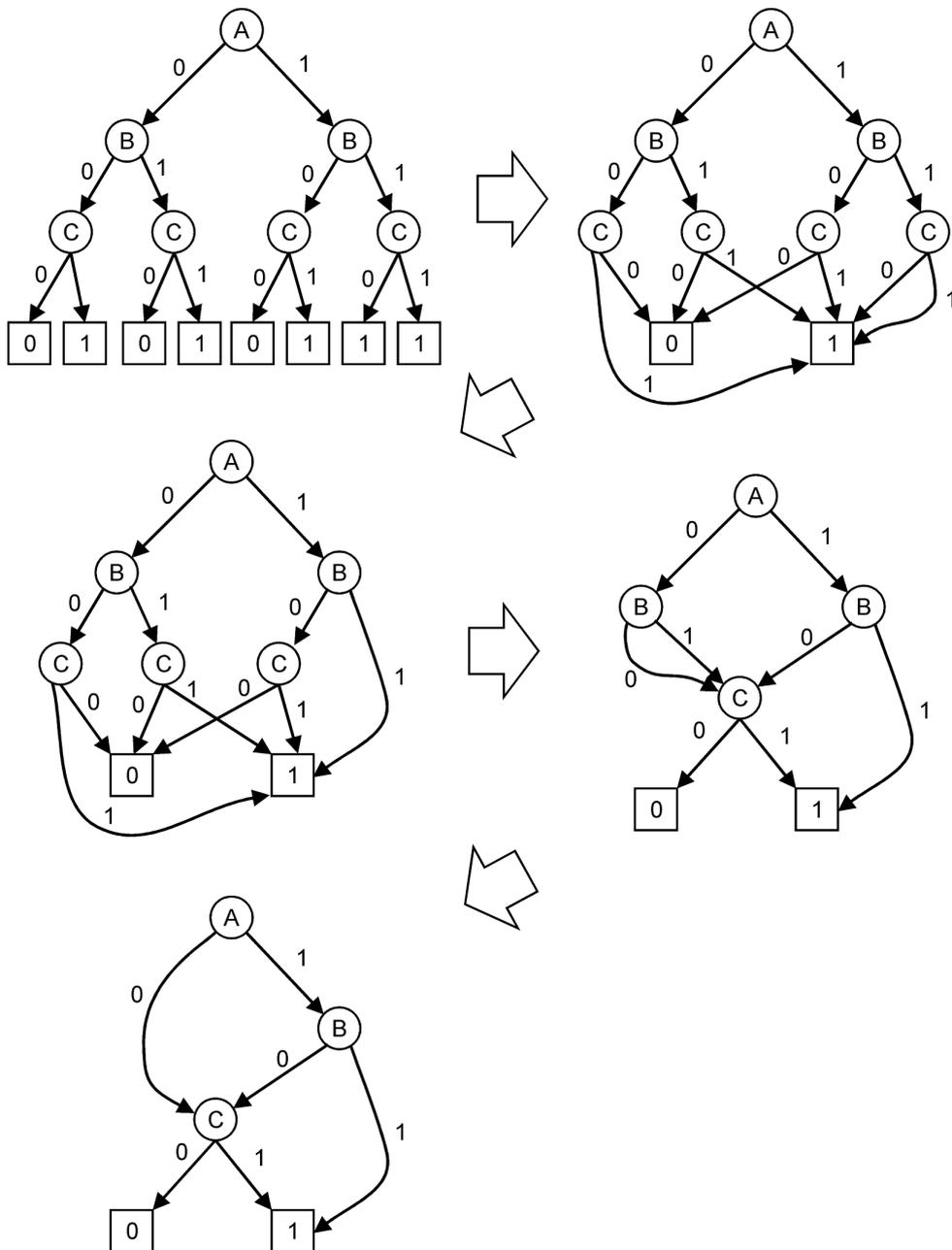
Aufgabe 5. (8 Punkte)

Gegeben sei die Boolesche Funktion f durch folgende Wahrheitstafel:

A	B	C	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Geben Sie das zu f gehörende BDD an, wenn die Variablenreihenfolge $A \rightarrow B \rightarrow C$ ist. Die Zwischenschritte müssen nachvollziehbar sein!

Lösung:



Aufgabe 6. (8 Punkte)

Gegeben sei die Funktion $f : \{0, 1\}^* \rightarrow \{0, 1\}$, mit:

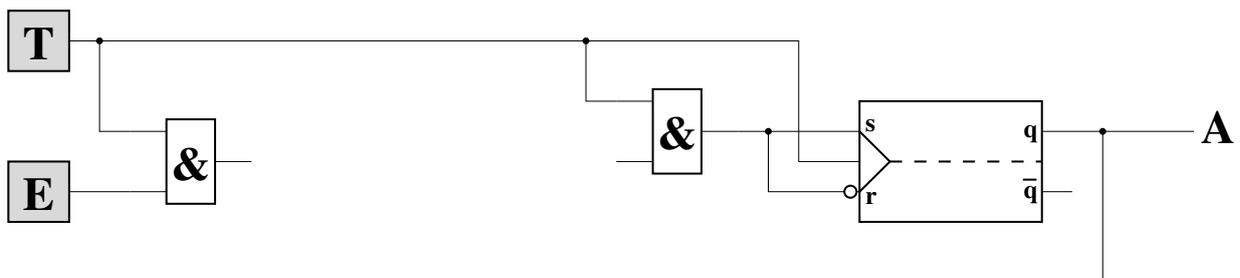
$$f(w) = \begin{cases} 0, & \text{falls die Summe der Einsen in } w \text{ geradzahlig ist,} \\ 1 & \text{sonst} \end{cases}$$

Entwerfen Sie ein Schaltwerk, das diese Funktion realisiert: Bei serieller Eingabe eines (beliebig langen!) Wortes soll auf einer Ausgabelitung zu jedem Zeitpunkt der Funktionswert $f(w')$ des bisher eingegebenen Teilwortes w' ausgegeben werden.

Erstellen Sie (analog zum Verfahren aus Vorlesung und Übung) eine Schalttabelle und vervollständigen Sie das Schaltbild. Verwenden Sie dazu (zusätzlich zu den bereits vorhandenen Gattern) nur UND-, ODER- und NICHT-Gatter.

q	E	q'

(E = Eingang, T = Takt, A = Ausgang.)

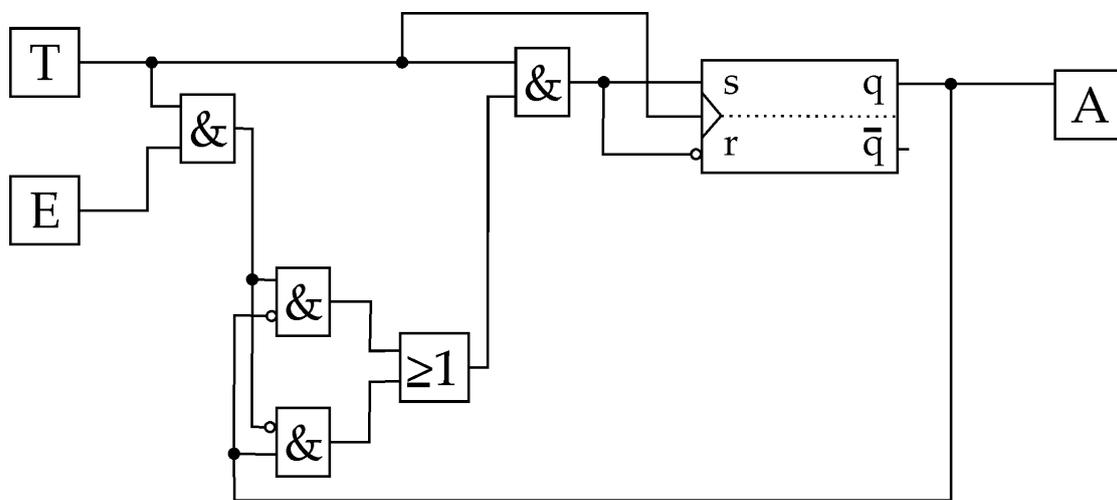


Lösung:

Es wird die Paritätsfunktion berechnet. Die Schalttabelle ist also:

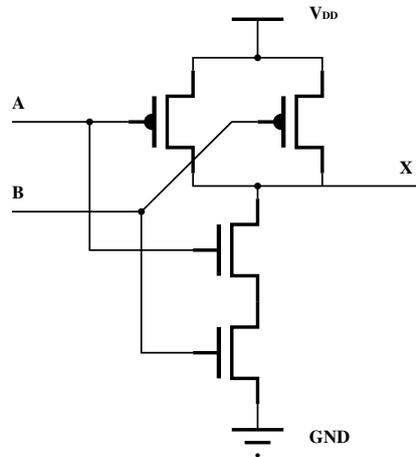
q	E	q'
0	0	0
0	1	1
1	0	1
1	1	0

Es gilt: $q'(q, E) = \bar{q}E + q\bar{E}$. Mit der entsprechenden Schaltfunktion muss das Flipflop angesteuert werden.



Aufgabe 7. (2 + 4 = 6 Punkte)

Die folgende Schaltung repräsentiert ein NAND-Gatter in CMOS-Darstellung.



- (a) Warum werden die PMOS-Transistoren mit VDD verbunden und die NMOS-Transistoren mit GND?

PMOS-Transistoren können gut eine logische 1 übertragen und NMOS-Trans. gut die logische 0.

- (b) Die Transistoren sind im P-Zweig parallel geschaltet und im N-Zweig in Reihe. Was würde passieren, wenn man bei beiden eine Reihenschaltung verwenden würde? Und bei Parallelschaltung?

Hinweis: unterscheiden Sie die Fälle $A = B$ und $A \neq B$.

Reihenschaltung:

Bei $A = B$ funktioniert die Schaltung weiterhin als Inverter.

$A \neq B$: In beiden Zweigen leitet ein Transistor und der andere sperrt. Wegen der Reihenschaltung gibt es kein definiertes Potential für X.

Parallelschaltung:

$A = B$: siehe Reihenschaltung.

$A \neq B$: In beiden Zweigen leitet ein Transistor und der andere sperrt. Wegen der Parallelschaltung gibt es einen geringen Widerstand zwischen VDD und GND, es fließt ein hoher Strom, der das Bauelement zerstören könnte. X liegt irgendwo zwischen VDD und GND.

Aufgabe 8. (7 + 3 = 10 Punkte)

Folgende Zeichenkette sei repräsentativ für Daten, die noch kommen sollen (Leerzeichen werden nicht codiert):

BLAUKRAUT BLEIBT BRAUTKLEID

- (a) Erzeugen Sie anhand der durch die Zeichenkette gegebenen Häufigkeitsverteilung eine Huffman-Codierung.

Tragen Sie dazu die Häufigkeiten der Zeichen in die erste Tabelle ein, erstellen Sie einen entsprechenden Baum mit Angabe der Wahrscheinlichkeiten an den Knoten und tragen Sie die Codierung der Zeichen in die zweite Tabelle ein.

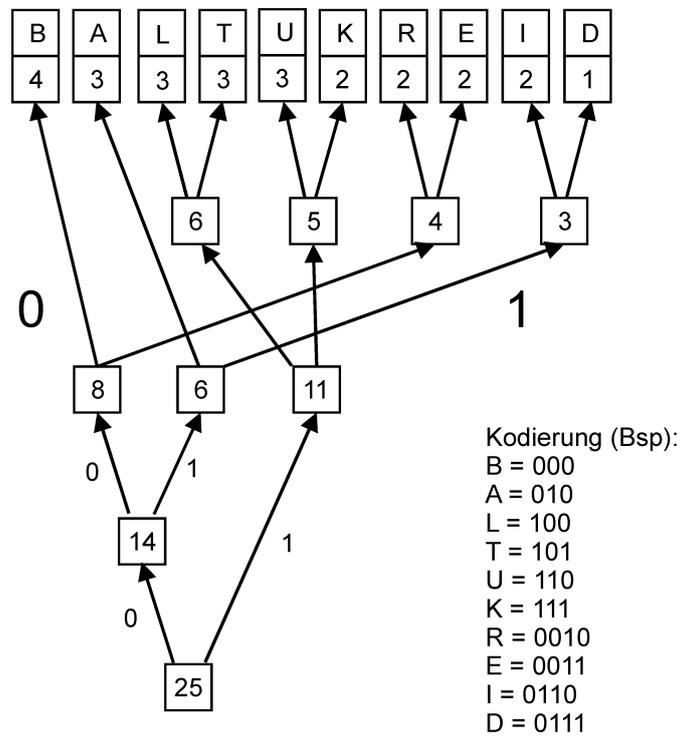
Zeichen	B	A	L	T	U	K	R	E	I	D
Häufigkeit										

Zeichen	Code
B	
A	
L	
T	
U	
K	
R	
E	
I	
D	

- (b) Geben Sie die Codelänge dieser Huffman-Codierung an (Bruch genügt).

Lösung (a):

B	4
A	3
L	3
T	3
U	3
K	2
R	2
E	2
I	2
D	1



Lösung (b):

$\frac{82}{25}$

Aufgabe 9. (1 + 2 + 4 = 7 Punkte)

Die folgende Liste zeigt einen Auszug aus dem ASCII-Code:

a: 1100001

b: 1100010

c: 1100011

d: 1100100

e: 1100101

f: 1100110

- (a) Bestimmen Sie die Hammingzahl des Teilcodes

Lösung: Hammingzahl: 1

- (b) Erweitern Sie den Code auf möglichst einfache Art, sodass 1-Bit-Fehler erkannt werden können.

a: 1100001-0

b: 1100010-0

c: 1100011-1

Lösung: d: 1100100-0

e: 1100101-1

f: 1100110-1

z.B. ein Paritätsbit anhängen.

(c) Ein anderer Code sieht folgendermassen aus:

a: 100001000000

b: 000000010001

c: 001000100000

d: 000100000010

e: 010000001000

f: 000010000100

Welche fehlerbehandelnden Eigenschaften hat dieser Code?

Lösung:

Hammingzahl: 4

Der Code kann 3-Bit-Fehler erkennen und 1-Bit-Fehler korrigieren

Aufgabe 10. (1 + 1,5 + 1,5 + 1 + 1 + 3 = 9 Punkte)

Gegeben sei die folgende Binärzahl:

$$b = 1000\ 1000\ 0000\ 0000\ 0000\ 0000\ 1000\ 0000$$

Geben Sie die zugehörige Dezimalzahl an, wenn b nach den folgenden Möglichkeiten codiert ist.

Es reicht aus, wenn Sie Summen von Zweierpotenzen schreiben, Sie müssen die Werte nicht ausrechnen.

(a) Zahl zur Basis 2

$$2^{31} + 2^{27} + 2^7$$

(b) 1-Komplement

$$-2^{31} + 2^{27} + 2^7 + 1$$

(c) 2-Komplement

$$-2^{31} + 2^{27} + 2^7$$

(d) Vorzeichen-Betrag

$$-(2^{27} + 2^7)$$

(e) Binary Coded Decimal (BCD)

$$88000080$$

(f) Gleitpunktdarstellung nach IEEE 754

$$n = 32, \quad k = 24, \quad q = 2^{8-1} - 1 = 127$$

$$v = 1, \quad c = 00010000_2 = 16_{10}, \quad m' = 2^{-16}$$

$$(-1)^1 \cdot (1 + 2^{-16}) \cdot 2^{16-127} = -(1 + 2^{-16}) \cdot 2^{-111} = \frac{2^{-16} + 1}{2^{111}}$$

Aufgabe 11. (4 Punkte)

Gegeben sei der folgende Ausschnitt eines Assemblerprogramms. Zu Beginn stehe in Register R1 ein Wert $n > 0$.

Die Befehle sind folgendermaßen aufgebaut: *OpCode* Q1, (Q2,) Z

(dabei steht 'Q' für Quelle und 'Z' für Ziel).

Für unmittelbare Adressierung wird der Präfix '#' verwendet.

JNZ Q L (JumpNotZero) ist ein bedingter Sprungbefehl, der zu Label *L* springt, falls $Q \neq 0$ gilt.

```
    STORE R1, R2
    STORE #0, R3
L1:  ADD R3, R2, R3
    SUBTRACT R1, #1, R1
    JNZ R1, L1
    STORE R3, R2
```

Wie hängt der Endwert von R2 mit dem Anfangswert von R1 zusammen?

Lösung:

$$R2 = n^2 (= R1^2).$$

Aufgabe 12. (6 Punkte)

Kreuzen Sie an, ob die folgenden Aussagen wahr oder falsch sind. Für jede richtige Antwort erhalten Sie einen Punkt, für falsche Kreuze wird Ihnen jeweils ein Punkt abgezogen. Sie können jedoch nicht weniger als 0 Punkte in dieser Aufgabe erzielen.

	Wahr	Falsch
Die Sprache $L = \{w \in \{a, b\}^* \mid \exists n \in \mathbb{N} : w = a^n b^n\}$ ist kontextsensitiv.	<input type="radio"/>	<input type="radio"/>
Die Äquivalenz zweier endlicher Automaten ist entscheidbar.	<input type="radio"/>	<input type="radio"/>
Es gibt kein Programm, für das entscheidbar ist, ob es nach endlicher Zeit anhält oder nicht.	<input type="radio"/>	<input type="radio"/>
Ein RS-Master-Slave-Flipflop entspricht 2 hintereinander geschalteten synchronen RS-Flipflops mit demselben Takt t , wobei t in negierter Form in das zweite Flipflop („Slave“) eingeht.	<input type="radio"/>	<input type="radio"/>
PROLOG ist eine einfache Maschinsprache.	<input type="radio"/>	<input type="radio"/>
Preemptive Scheduling heißt, dass ein Prozess vor seiner Fertigstellung unterbrochen werden kann.	<input type="radio"/>	<input type="radio"/>

Lösung

	Wahr	Falsch
Die Sprache $L = \{w \in \{a, b\}^* \mid \exists n \in \mathbb{N} : w = a^n b^n\}$ ist kontextsensitiv.	<input checked="" type="radio"/>	<input type="radio"/>
Die Äquivalenz zweier endlicher Automaten ist entscheidbar.	<input checked="" type="radio"/>	<input type="radio"/>
Es gibt kein Programm, für das entscheidbar ist, ob es nach endlicher Zeit anhält oder nicht.	<input type="radio"/>	<input checked="" type="radio"/>
Ein RS-Master-Slave-Flipflop entspricht 2 hintereinander geschalteten synchronen RS-Flipflops mit demselben Takt t , wobei t in negierter Form in das zweite Flipflop („Slave“) eingeht.	<input checked="" type="radio"/>	<input type="radio"/>
PROLOG ist eine einfache Maschinsprache.	<input type="radio"/>	<input checked="" type="radio"/>
„Preemptive Scheduling“ heißt, dass ein Prozess vor seiner Fertigstellung unterbrochen werden kann.	<input checked="" type="radio"/>	<input type="radio"/>