Lösungsvorschläge und Erläuterungen

Klausur zur Vorlesung
Grundbegriffe der Informatik (6 ECTS)
19. September 2017

Klausur-nummer				

Nachname:
Vorname:
Matr.-Nr.:

Diese Klausur ist mein ☐ 1. Versuch ☐ 2. Versuch in GBI

Email-Adr.:

nur falls 2. Versuch

Aufgabe	1	2	3	4	5	6	
max. Punkte	1	10	8	12	10	10	
tats. Punkte							

Gesamtpunktzahl:			Note:	

/ 1

Aufgabe 1 (1 Punkt)

Beschriften Sie die Titelseite mit Ihrem Namen, Ihrer Matrikelnummer sowie Ihrer Klausurnummer. Geben Sie an, ob dies Ihr erster oder zweiter Versuch ist. Falls dieses Ihr zweiter Versuch ist, geben Sie bitte eine E-Mail Adresse an, unter der wir Sie erreichen können.

Beschriften Sie jedes weitere Blatt in der Kopfzeile der Vorderseite mit Ihrem Namen und Ihrer Matrikelnummer.

/ 10	**Aufgabe 2** (1 + 1 + 1 + 1.5 + 0.5 + 1.5 + 2 + 1.5 = 10 Punkte)

a) Es sei $A = \{a, b, c\}$ und $B = \{0, 1\}$. Geben Sie in geeigneter Form einen Homomorphismus $h\colon A^* \to B^*$ an, der ϵ-frei ist. /1

$h(a) = 0$, $h(b) = 1$, $h(c) = 1$

b) Gegeben sei folgende aussagenlogische Formel $G = ((A \land B) \lor ((\neg A) \lor B))$. Ist die Interpretation I mit $I(A) = 1$ und $I(B) = 0$ ein Modell von G? Begründen Sie Ihre Antwort. /1

Nein, da $val_I(G) = f$

c) Sei $f\colon L_A \to L_B$ eine Übersetzung. Ferne gelte für alle $x_1, x_2 \in L_A$: $f(x_1) = f(x_2) \Rightarrow x_1 = x_2$. Ist f eine Codierung? Begründen Sie Ihre Antwort. /1

Ja, da f injektiv ist.

d) Listen Sie die in der Vorlesung eingeführten Einheiten für das Maß der Information einer Nachricht auf. /1.5

natural units (nat), hartley (Hart), Shannon (Sh) bzw. binary digits (bits)

e) Ist eine Schallwelle eine Nachricht? /0.5

Antwort: Nein

f) Welche der folgenden Mengen sind Alphabete? /1.5

1) $\{0, 1, 0, 1\}$
2) $\{x \in \mathbb{N} \mid x^2 \geq 100\}$
3) $\{x \in \mathbb{N} \mid 3 * x + 100 \leq 1000\}$
4) $\{\}$
5) $\{1, 2, 3\} \times \{a, b, c, d\}$
6) $\mathbb{N}$

1,3,5

/2

g) Welche der folgenden Aussagen zum größenordnungsmäßigen Wachstum für die Funktionen $f : \mathbb{N}_0 \to \mathbb{R}_0^+$ und $g : \mathbb{N}_0 \to \mathbb{R}_0^+$ sind richtig?

1) $f \asymp g$ für $f\colon n \mapsto 26n^4$ und $g\colon n \mapsto 12^{-3}n^4$
2) $f \asymp g$ für $f\colon n \mapsto n^2$ und $g\colon n \mapsto n^{2\log n}$
3) $f \asymp g$ für $f\colon n \mapsto 15n^4$ und $g\colon n \mapsto 3n^3$
4) $f \asymp g$ für $f\colon n \mapsto \ln n$ und $g\colon n \mapsto e^n$
5) $f \asymp g$ für $f\colon n \mapsto \log_{10} n$ und $g\colon n \mapsto \ln n$

1,5

/1.5

h) Welche Eigenschaften muss eine Relation aufweisen, damit sie eine Äquivalenzrelation ist?

Reflexivität, Symmetrie und Transitivität

Weiterer Platz für Antworten zu Aufgabe 2:

/ 8 **Aufgabe 3** **(3 + 2 + 3 = 8 Punkte)**

Gegeben seien die beiden Funktionen $f\colon \mathbb{N}_0 \to \mathbb{R}_0^+$ und $g\colon \mathbb{N}_0 \to \mathbb{R}_0^+$.

/3

a) Für $f\colon n \mapsto 2n^4 + 4n^3$ und $g\colon n \mapsto 5n^4 - 2n^2$ beweisen oder widerlegen Sie: $f \asymp g$

/2

b) Beweisen oder widerlegen Sie die folgende Behauptung: Für alle $f\colon \mathbb{N}_0 \to \mathbb{R}_0^+$ und jede Konstanten $a, b \in \mathbb{N}_0$ gilt: $\Theta(af) = \Theta(bf)$

Binäre-Suche ist ein Algorithmus zum Finden des Index `idx` eines Elements `x` in einer sortierten Liste der Länge `n` von `n` verschiedenen Zahlen. Der erste Index der Liste `liste` sei `1`. Eine Implementierung des Algorithmus ist hier in Pseudo-Code angegeben:

```
idx = binsearch(liste, 1, n)

funktion binsearch(liste, links, rechts, x):
  if (links <= rechts) then
      mitte = (links + rechts) / 2
      if (x == liste[mitte]) then
        return mitte
      elseif (x < liste[mitte]) then
        return binsearch(liste, links, mitte-1, x)
      else
        return binsearch(liste, mitte+1, rechts, x)
  else
      return 0
```

/3

c) Geben Sie die Laufzeit des Algorithmus in Abhängigkeit von der Länge n der Liste, in der gesucht wird, in Θ-Notation an. Begründen Sie Ihr Ergebnis.

Platz für Antworten zu Aufgabe 3:

Weiterer Platz für Antworten zu Aufgabe 3:

Weiterer Platz für Antworten zu Aufgabe 3:

Lösung 3

a) Behauptung: $f \asymp g$. Beweis:

- $$\text{für } n >= 0 \text{ ist:} \quad \begin{aligned} f(n) &= 2n^4 + 4n^3 \\ &\leq 2n^4 + 4n^4 \\ &= 6n^4 \\ &= 10n^4 - 4n^4 \\ &\leq 10n^4 - 4n^2 \\ &= 2(5n^4 - 2n^2) \quad = 2g(n) \end{aligned}$$
  $$\text{also} \quad \frac{1}{2} f(n) \leq g(n)$$
- $$\text{Andereseits gilt für } n \geq 0 \quad \begin{aligned} g(n) &= 5n^4 - 2n^2 \\ &\leq 5n^4 \\ &= \frac{5}{2} 2n^4 \\ &\leq \frac{5}{2}(2n^4 + 4n^3) = \frac{5}{2} f(n) \end{aligned}$$
  $$\text{also} \quad \frac{5}{2} f(n) \geq g(n)$$

b) Die Behauptung ist falsch. Beweis durch Gegenbeweis. Es sei $a = 0$, $b = 1$, $f\colon n \mapsto 1$. Dann gilt für alle $n \in \mathbb{N}_0$: $0 = af(n) < bf(n) = 1$. D.h. es gibt kein $c \in \mathbb{R}_+$ und kein $n_0 \in \mathbb{N}_0$, sodass für alle $n \geq n_0$ gilt, dass $caf(n) \geq bf(n)$. Damit gilt $bf \notin \Theta(af)$. Da aber $bf \in \Theta(bf)$ gilt somit $\Theta(af) \neq \Theta(bf)$.

c) Dies ist ein Beispiel eines Teile-und-Herrsche-Algorithmus. Das Problem wird jeweils in Probleme halber Größe geteilt. D.h. für die Laufzeit $T(n)$ gilt: $T(n) = 1T(n/2) + f(n)$. Die Kombination geschieht immer in $O(1)$ weil nur ein Vergleich `x = liste[mitte]` notwendig ist. Also $f(n) \in \Theta(n^0) = \Theta(n^{\log_2(1)})$. Die Anwendung des zweiten Falls des Mastertheorems ergibt somit $T(n) \in \Theta(\log n)$

/ 12

Aufgabe 4 **(1.5 + 2 + 5 + 2.5 + 1 = 12 Punkte)**

Gegeben Sei folgende Turingmaschine $T_1 = (Z, z_0, X, f, g, m)$ mit

- Zustandsmenge $Z = \{1, 2\}$
- Anfangszustand $z_0 = 1$
- Bandalphabet $X = \{a, b, \square\}$
- Zustandsüberführungsfunktion

$$
\begin{aligned}
f\colon Z \times X &\to Z \text{ mit,}\\
(1, a) &\mapsto 1\\
(2, a) &\mapsto 2\\
(1, b) &\mapsto 1\\
(2, b) &\mapsto 2\\
(1, \square) &\mapsto 1\\
(2, \square) &\mapsto 1
\end{aligned}
$$

- Ausgabefunktion

$$
\begin{aligned}
g\colon Z \times X &\to X \text{ mit,}\\
(1, a) &\mapsto a\\
(2, a) &\mapsto \square\\
(1, b) &\mapsto a\\
(2, b) &\mapsto \square\\
(1, \square) &\mapsto a\\
(2, \square) &\mapsto a
\end{aligned}
$$

- Bewegungsfunktion

$$
\begin{aligned}
m\colon Z \times X &\to \{-1, 0, 1\} \text{ mit,}\\
(1, a) &\mapsto -1\\
(2, a) &\mapsto 1\\
(1, b) &\mapsto -1\\
(2, b) &\mapsto 1\\
(1, \square) &\mapsto 1\\
(2, \square) &\mapsto -1
\end{aligned}
$$

a) Hält T_1 für jede beliebige Anfangskonfiguration c? Begründen Sie Ihre Antwort. /1.5

Lösung 4

Nein

- Zustandsüberführungs-, Ausgabe- und Bewegungsfunktion sind vollständig defininiert (oder)
- Durch Beispiel: Für leeres Band und den Anfangszustand $z_0 = 1$ hält die Turingmaschine nie

Für T_1 sein nun zusätzlich folgende Konfiguration $c = (z_0, b, p) \in Z \times X^{\mathbb{Z}} \times \mathbb{Z}$ gegeben mit:

- $b \colon \mathbb{Z} \to X$ mit

$$n \mapsto \begin{cases} b, & \text{für z=-1,} \\ a, & \text{für z=0,} \\ a, & \text{für z=1,} \\ \square, & \text{sonst} \end{cases}$$

- $p = 0$

b) Geben Sie die beiden Folgekonfigurationen $\Delta_1(c)$ und $\Delta_2(c)$ für T_1 an. /2

Lösung 4

Die jeweilige Position des Kopfes der Turingmaschine ist fett gedruckt.

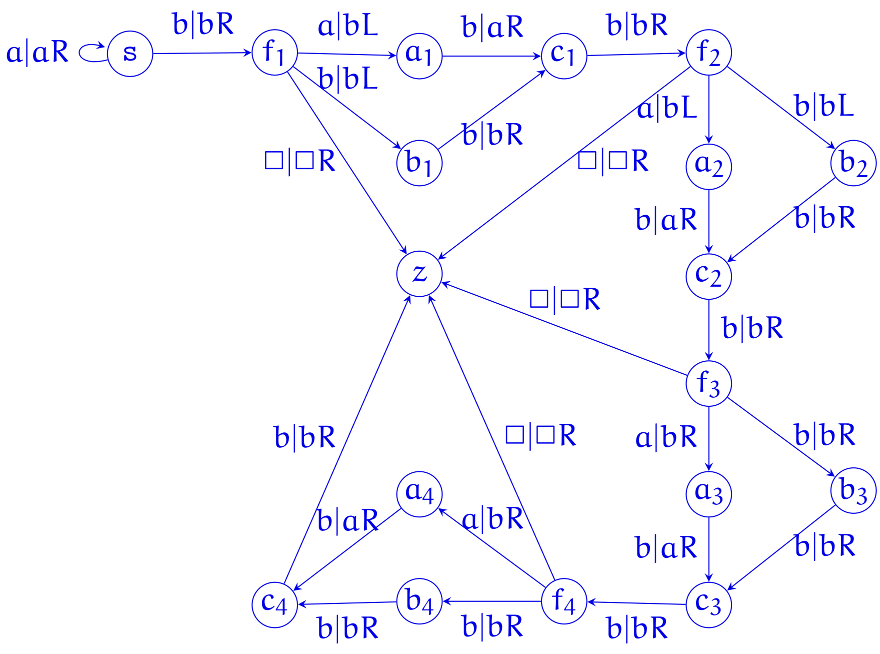
	-3	-2	-1	0	1	2	3	
$\Delta_0:$	$\square$	$\square$	b	$\mathbf{a}$	a	$\square$	$\square$	Zustand : 1
$\Delta_1:$	$\square$	$\square$	$\mathbf{b}$	a	a	$\square$	$\square$	Zustand : 1
$\Delta_2:$	$\square$	$\boldsymbol{\square}$	a	a	a	$\square$	$\square$	Zustand : 1

c) Geben Sie graphisch eine Turingmaschine T_2 an, die wie folgt operiert. Die Eingabe der Turingmaschine bestehe dabei aus einem Wort $w \in \{a, b\}^+$, das zu Anfang auf dem Band steht. Sonst stehen auf dem Band nur Blanksymbole. Der Kopf der Turingmaschine stehe am Anfang auf dem ersten Zeichen des Wortes w. Die Turingmaschine soll nun das Wort von links nach rechts durchgehen und das erste vorkommende Zeichen b im Wort um vier Positionen nach rechts im Wort verschieben oder, falls das Teilwort nach dem ersten Vorkommen von b dafür nicht lang genug ist, an das Ende des Wortes stellen. Es sollen keine Zeichen des Wortes beim Verschieben des b überschrieben werden, sondern es soll durch entsprechendes Verschieben der Zeichen Platz für das b an seiner neuen Position geschaffen werden. Auch soll es keine Lücke im Wort geben, sondern die Lücke, die durch das Verschiebenen des b entsteht, soll durch Verschieben der anderen Zeichen entsprechend geschlossen werden. So soll z.B. $w = abaaaaa$ nach $w' = aaaaaba$ überführt werden und $w = ba$ nach $w' = ab$. /5

Lösung 4

Zustände (Index $i \in \{1, 2, 3, 4\}$ zählt die Anzahl der Verschiebungen):

- s: Startzustand
- z: Endzustand
- f_i: links vom Kopf steht ein b (found b)
- a_i: a gemerkt, rechts vom Kopf steht ein b
- b_i: b gemerkt, rechts vom Kopf steht ein b
- c_i: gehe einen Schritt weiter (continue),



/2.5

d) Vervollständigen Sie in der Vorlage unten graphisch die Anfangskonfiguration Δ_0 Ihrer Turingmaschine T_2 für das Wort $w = abaaaaa$ und geben Sie weiterhin graphisch in der unten stehenden Vorlage die ersten vier Folgekonfigurationen für w an.

Δ_0:	$\square$	a	b	a	a	a	a	a	$\square$	$\square$	$\square$	$\square$	$\square$	$\square$

/1

e) Geben Sie für Ihre Turing-Maschine T_2 möglichst scharf $\mathrm{Time}_T(n)$ in O-Notation an, wobei n die Länge des Eingabewortes ist.

Lösung 4

$\mathcal{O}(n)$

Weiterer Platz für Antworten zu Aufgabe 4:

Weiterer Platz für Antworten zu Aufgabe 4:

Lösung 4

Die jeweilige Position des Kopfes der Turingmaschine ist fett gedruckt.

Δ_0:	$\square$	$\mathbf{a}$	b	a	a	a	a	a	$\square$	$\square$	$\square$	Zustand : s
Δ_1:	$\square$	a	$\mathbf{b}$	a	a	a	a	a	$\square$	$\square$	$\square$	Zustand : s
Δ_2:	$\square$	a	b	$\mathbf{a}$	a	a	a	a	$\square$	$\square$	$\square$	Zustand : f_1
Δ_3:	$\square$	a	$\mathbf{b}$	b	a	a	a	a	$\square$	$\square$	$\square$	Zustand : a_1
Δ_4:	$\square$	a	a	$\mathbf{b}$	a	a	a	a	$\square$	$\square$	$\square$	Zustand : c_1

/ 10

Aufgabe 5 (2 + 1 + 2 + 5 = 10 Punkte)

Gegeben sei der reguläre Ausdruck: $R_1 = (\texttt{a} \mid \texttt{b}) * \texttt{b}(\texttt{a} \mid \emptyset)((\texttt{a} \mid \texttt{b}) \mid \texttt{c})(\texttt{a} \mid \texttt{b})*$

a) Geben Sie den Ableitungsbaum des regulären Ausdrucks R_1 gemäß der Grammatik für reguläre Ausdrücke aus der Vorlesung an. /2

b) Zeichnen Sie den Kantorowitsch-Baum für den regulären Ausdruck R_1. /1

c) Geben Sie eine rechtslineare Grammatik G_r an, so dass $L(G_r) = \langle R_1 \rangle$. /2

Das Pumping-Lemma für reguläre Sprachen lautet wie folgt: Für jede reguläre Sprache L gibt es ein $n \in \mathbb{N}$, also insbesondere $n \geq 1$, so dass sich jedes Wort $z \in L$ mit $|z| \geq n$ zerlegen läßt in $z = uvw$ mit

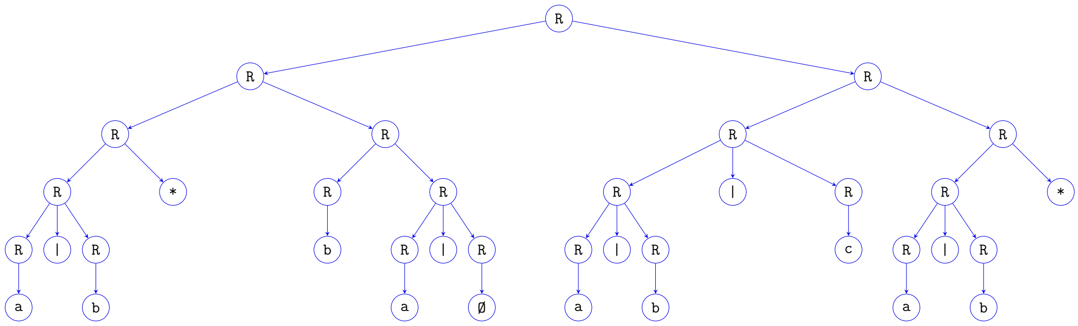
- $|uv| \leq n$
- $|v| \geq 1$
- Für jedes $i \in \mathbb{N}_0$ ist das Wort $uv^i w \in L$

d) Zeigen Sie mit Hilfe des Pumping-Lemmas, dass die Sprache $L = \{w \in \{0, 1\}^* \mid w \text{ ist ein Palindrom}\}$ nicht regulär ist. Überlegen Sie dazu, welche Wörter gemäß des Pumping-Lemmas in L liegen müssten, wenn L eine reguläre Sprache wäre. /5

Platz für Antworten zu Aufgabe 5:

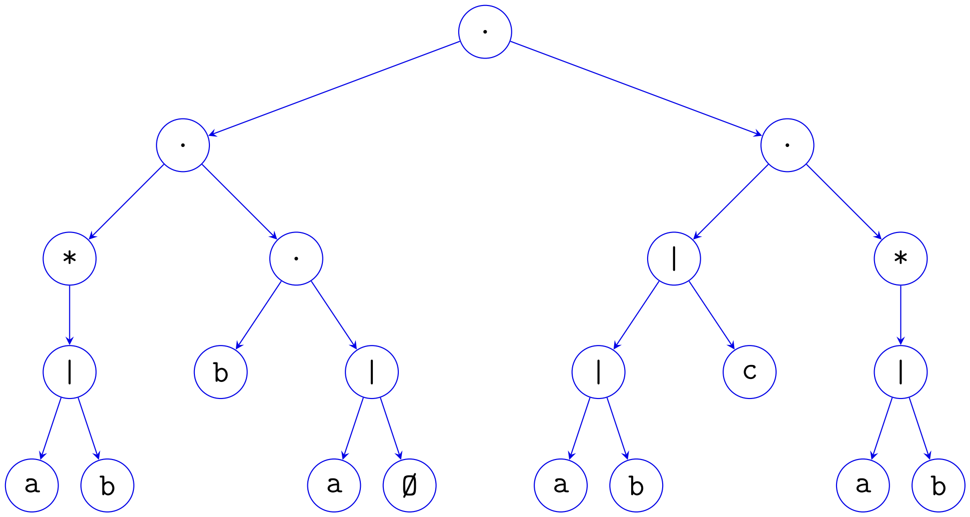
Lösung 5

Ableitungsbaum:
Zur Übersichtlichkeit wurden die Klammern weggelassen



Lösung 5

Kantorowitsch-Baum



Weiterer Platz für Antworten zu Aufgabe 5:

Lösung 5

Grammatik: $G = (N, T, S, P)$ mit:

$$
\begin{aligned}
N &= \{S, A, B, C, D\},\\
T &= \{a, b, c\},\\
P &= \{\\
&\quad S \rightarrow aS|bS|A,\\
&\quad A \rightarrow bB,\\
&\quad B \rightarrow aC|C,\\
&\quad C \rightarrow aD|bD|cD,\\
&\quad D \rightarrow aD|bD|\varepsilon\\
&\quad \}
\end{aligned}
$$

Weiterer Platz für Antworten zu Aufgabe 5:

Lösung 5

Wähle das Palindrom $z = 0^n10^n \in L$ mit beliebigen, aber fest gewähltem n. Damit lässt sich das Pumping-Lemma anwenden und wir zerlegen z in $z = uvw$ mit $|uv| \leq n$ und $|v| \geq 1$. Daraus folgt:

- $uv \in 0^*$
- $v \in 0^*$
- $w \in 0^*10^n$
- Da $|v| \geq 1$ besteht v aus mindestens einer 0

Demnach ist $uv^0w = uw = 0^k10^n$ mit $k < n$ offensichtlich kein Palindrom. Folglich ist $uv^0w \notin L$, was ein Widerspruch zum Pumping-Lemma ist. Deshalb ist L keine reguläre Sprache.

/ 10

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

Als Erinnerungshilfe finden Sie hier eine Liste der MIMA-Befehle, die Sie in der Vorlesung kennengelernt haben:

`LDC` *const*
`LDV` *adr*
`STV` *adr*
`LDIV` *adr*
`STIV` *adr*
`ADD` *adr*
`AND` *adr*
`OR` *adr*
`XOR` *adr*
`NOT`
`RAR`
`EQL` *adr*
`JMP` *adr*
`JMN` *adr*

/3

a) Schreiben Sie ein MIMA-Programm, dass zwei natürliche Zahlen a und b vergleicht und das höchstwertige Bit des Akkus auf 1 setzt falls $a > b$ und auf 0 sonst. Schreiben Sie Ihr Programm unter Verwendung der in der Vorlesung eingeführten MIMA-Befehle in die Vorlage unten. Verwenden Sie nicht mehr Befehle als in der Vorlage vorgesehen sind. Neben den Speicheradressen aus der Vorlage steht Ihnen noch eine Speicheradresse `tmp` zur Verfügung, die Sie für Ihr Programm verwenden dürfen. Die Zahl a liegt im Speicher an der Adresse `a_adr` , die Zahl b liegt an der Adresse `b_adr` .

Speicheradresse	Befehl
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	

b) Nehmen Sie an, dass Sie jetzt einen MIMA-Befehl `MAX` *adr* zur Verfügung haben, der den Wert im Akku mit dem Wert an der Adresse `adr` vergleicht und das höchstwertige Bit des Akkus auf 1 setzt, falls der Wert im Akku größer ist als der Wert an der Speicheradresse `adr` , und auf 0 sonst. Schreiben Sie jetzt in die Vorlage unten ein MIMA-Program, dass das Maximum dreier natürlicher Zahlen a, b und c bestimmt, dieses im Akku ablegt und danach die Ausführung an der Speicheradresse `continue` fortsetzt. Neben den Speicheradressen aus der Vorlage steht Ihnen noch eine Speicheradresse `tmp` zur Verfügung, die Sie für Ihr Programm verwenden dürfen. Verwenden Sie nicht mehr Befehle als in der Vorlage vorgesehen sind. Die Zahlen a, b und c liegen an den Speicheradressen `a_adr` , `b_adr` und `c_adr` .

/6

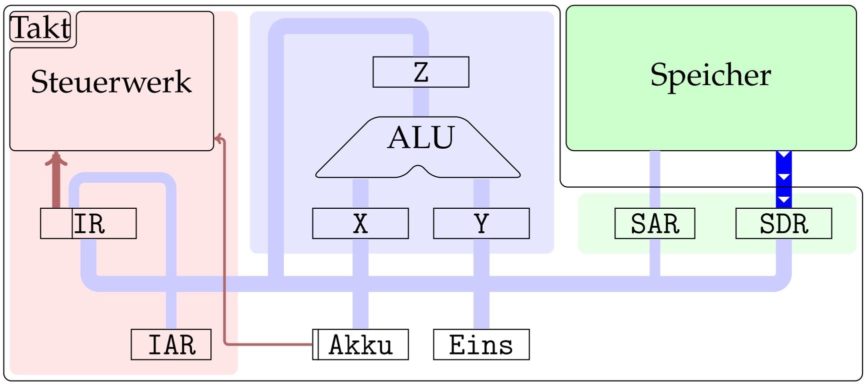
Speicheradresse	Befehl
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	

15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	

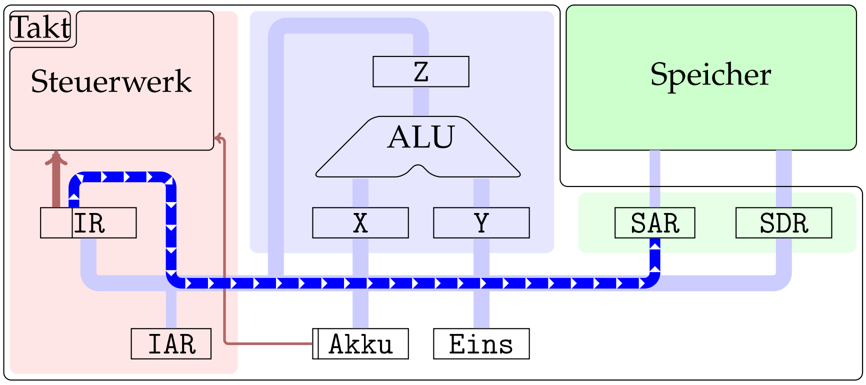
c) Folgende vier Bilder zeigen unterschiedliche Schritte der Befehlsausführungsphase für `LDV` *adr*.

/1

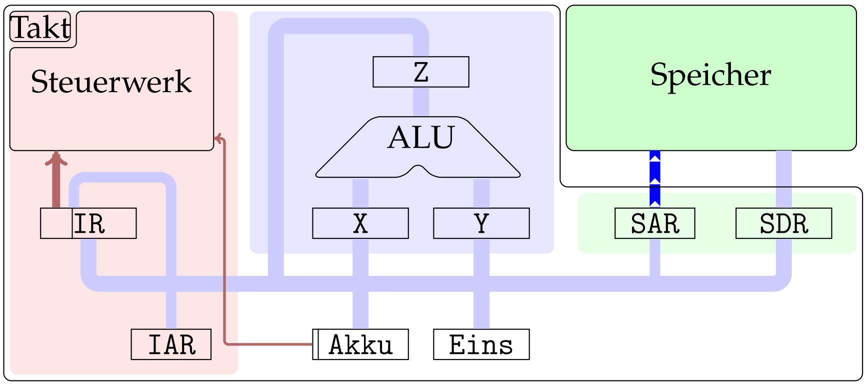
1.



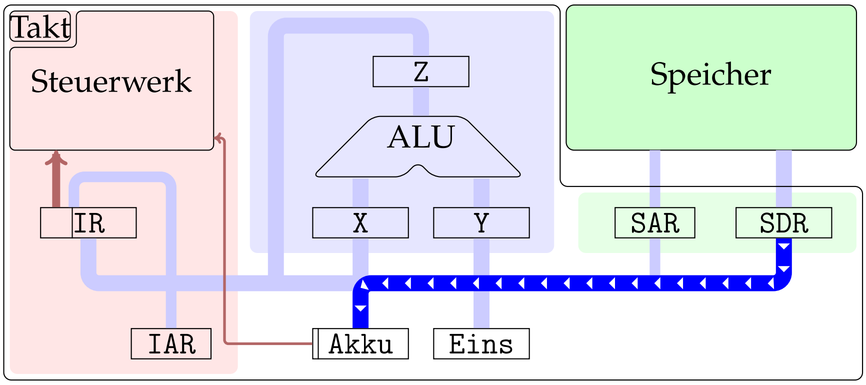
2.



3.



4.



Bringen Sie die vier Bilder in die richtige Reihenfolge.

2, 3, 1, 4

Weiterer Platz für Antworten zu Aufgabe 6:

Weiterer Platz für Antworten zu Aufgabe 6:

Lösung 6

a)

Speicheradresse	Befehl
0	`LDV` *a_adr*
1	`NOT`
2	`STV` *tmp*
3	`LDC` *1*
4	`ADD` *tmp*
5	`ADD` *b_adr*

b)

Speicheradresse	Befehl
0	`LDV` *a_adr*
1	`MAX` *b_adr*
2	`JMN` *10*
3	`LDV` *b_adr*
4	`MAX` *c_adr*
5	`JMN` *8*
6	`LDV` *c_adr*
7	`JMP` *CONTINUE*
8	`LDV` *b_adr*
9	`JMP` *CONTINUE*
10	`LDV` *a_adr*
11	`MAX` *c_adr*
12	`JMN` *15*
13	`LDV` *c_adr*
14	`JMP` *CONTINUE*
15	`LDV` *a_adr*
16	`JMP` *CONTINUE*

oder alternativ:

Speicheradresse	Befehl
0	`LDV` *a_adr*
1	`MAX` *b_adr*
2	`JMN` *5*
3	`LDV` *b_adr*
4	`JMP` *6*
5	`LDV` *a_adr*
6	`STV` *tmp*
7	`LDV` *c_adr*
8	`MAX` *tmp*
9	`JMN` *12*
10	`LDV` *tmp*
11	`JMP` *CONTINUE*
12	`LDV` *c_adr*
13	`JMP` *CONTINUE*