

2. Klausur 11/II (EF.2) (N)

Dauer: 180 Minuten

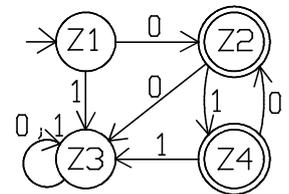
Name: www.r-krell.de

Hilfsmittel: normaler Taschenrechner

- * *Achte auf sorgfältige Darstellung mit vollständigem, nachvollziehbarem Lösungsweg!* *
- * *Kommentiere deine Programme!* *

- ① Gegeben ist der erkennende Automat M durch $M = \{\mathcal{E}, Z, \delta, Q, \{P\}\}$, wobei $\mathcal{E} = \{a, b\}$ und $Z = \{P, Q, R\}$ ist. Die Übergangsfunktion $\delta: Z \times \mathcal{E} \rightarrow Z$ ist durch vollständige Aufzählung festgelegt: $\delta(Q,a)=Q, \delta(Q,b)=R, \delta(P,a)=Q, \delta(P,b)=P, \delta(R,a)=Q$ und $\delta(R,b)=P$.
- a) Zeichne den Automatengraph
 - b) Prüfe folgende Wörter und notiere, in welchem Zustand sie enden und ob sie vom Automaten akzeptiert werden oder nicht: (1) aba, (2) bbaa, (3) aabb und (4) ababba.
 - c) Gib selbst noch je zwei eigene Beispiel-Wörter an, die akzeptiert werden, sowie zwei Beispielwörter, die nicht akzeptiert werden. Beschreibe außerdem kurz in Deutsch, welche Eigenschaft akzeptierte Wörter haben müssen.

- ② Gegeben ist der nebenstehende Graph eines erkennenden Automaten, der eingegebene Dualzahlen („Wörter“) genau dann akzeptiert, wenn sie nicht negativ sind und nie zwei gleiche Ziffern aufeinander folgen.



- a) Notiere die formale Beschreibung $A = \{\mathcal{E}, Z, \delta, \dots\}$ des Automaten, wobei die Übergangsfunktion δ durch eine Automatentafel festgelegt werden soll.
- b) Schreibe eine Java-Methode *aufgabe2b*, die den Automaten schematisch nachbildet und genau dann *true* liefert, wenn das als String übergebene Wort akzeptiert wird.. (Hinweis: "abcdefg".charAt(1) = 'b', weil die Nummerierung mit 0 beginnt).
- c) Schreibe eine eigene Java-Methode *aufgabe2c*, die (ggf. mehrfach) den Befehl *indexOf* verwendet und genau dann *true* liefert, wenn das als String übergebene Wort mit "0" beginnt und nie zwei gleiche Ziffern direkt hintereinander enthält, also akzeptiert wird. (Hinweis: "abcdefg".indexOf("cde") liefert 2, weil der Teilstring an der 2. Stelle von "abcdefg" beginnt, während "abc".indexOf("xy") = -1 ist, weil "xy" gar nicht in "abc" vorkommt.)
- d) Schreibe ein 1_AMOR-Maschinenprogramm, das genau dann eine 1 in Zelle 62 ausgibt, wenn die zehnstellige Dualzahl in Zelle 63 akzeptiert wird. Sonst soll in Zelle 62 eine 0 bleiben. Beschreibe anfangs kurz die Idee und kommentiere dein Maschinenprogramm!
- e) Erläutere, warum es nicht nötig ist, eine Wahrheitstafel und ein KV-Diagramm zu erstellen, wenn man eine Hardwareschaltung angeben will, die bei Eingabe von 4-stelligen Dualzahlen genau dann eine 1 ausgibt, wenn die Dualzahl akzeptiert wird. Zeichne die Hardware!
- f) Notiere eine (Chomsky-Typ3-)Grammatik für die Sprache, die aus allen akzeptierten Wörtern besteht. Nenne dazu die Menge \mathbb{N} der Nichtterminale, die Menge \mathbb{T} der Terminale, alle benötigten Regeln [Produktionen] (einzeln oder in BNF) und das Startsymbol.

- ③ Zeichne den Automatengraphen (oder gib statt dessen die Automatentafel an) für einen erkennenden Automaten, der Wörter aus den Buchstaben a und b genau dann akzeptiert, wenn das Wort mindestens ein a und eine ungerade Anzahl von b enthält. (Beisp.: Akzeptiert werden z.B. ab oder baaabb, aber nicht abb oder bbb).

- ④ Ein 1_AMOR-Maschinenprogramm soll die positive Zahl in Zelle 62 auf folgende Weise verarbeiten und das Ergebnis wieder in Zelle 62 ausgeben: Ist die Zahl größer 15, dann soll ausgegeben werden, wie viel die Zahl größer als 15 ist (Beisp: Eingabe 38 führt zur Ausgabe 23, weil 38 um 23 größer ist als 15). Kleinere Zahlen sollen zur Ausgabe von -1 führen.
- a) Schreibe das ganze entsprechende 1_AMOR-Programm mit einem SUB(traktions)-Befehl (kommentiert)
 - b) Jetzt soll die Subtraktion „Zahl minus 15“ nicht mehr verwendet werden. Statt dessen soll die Eingabezahl um eine feste negative Anzahl von Stellen nach links (= um eine positive Abzahl

- von Stellen nach rechts) verschoben werde und damit ermittelt werden, ob die Zahl > 15 ist/war. Erläutere, warum das geht und schreibe die notwendigen Änderungen des Programms!
- c) In deinen Programmen a) und/oder b) kommen SP< oder SP= vor. Beschreibe genau Hol- und Ausführungsphase eines solchen Befehls (für den Fall, dass gesprungen wird) und begründe, warum der 4. Schritt der Holphase nicht der erste sein darf!

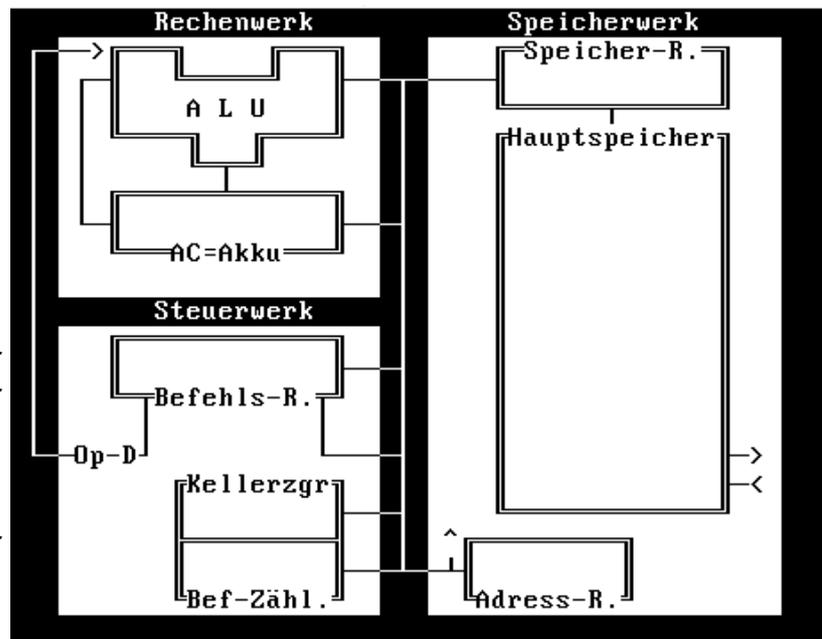
Modellrechner 1_AMOR -- Architektur und Befehlssatz

Architektur (Aufbau) des Modellrechners 1_AMOR

Der Hauptspeicher besteht aus 64 Zellen à 10 Bit (10-Bit-Speicherworte), nummeriert von 0 bis 63.

Z. 62: Bildschirmspeicher
Z. 63: Tastaturpuffer

(Das abgebildete Kellerzeiger-Register wird in dieser Klausur nicht gebraucht)



dual-dez--mnem.----Beschreibung des Maschinenbefehls *Java + goto*

0000	0)	LAD	n	Lade (aus dem Speicher in den Akku) $ac = mem[n];$
0001	1)	TRF	n	Transferiere (Speichere) $mem[n] = ac;$
0010	2)	NOP	n	(no operation)
0011	3)	NOP	n	(no operation)
0100	4)	AND	n	Verknüpfe bitweise mit "Und" $ac = ac \& mem[n];$
0101	5)	OR	n	Verknüpfe bitweise mit "Oder" $ac = ac \mid mem[n];$
0110	6)	NOT	n	Invertiere bitweise $ac = \sim ac; // = -1-ac$
0111	7)	ADD	n	Addiere $ac = ac + mem[n];$
1000	8)	SUB	n	Subtrahiere $ac = ac - mem[n];$
1001	9)	SHL	n	Schiebe links um mem[n] Stellen $ac = ac \ll mem[n];$
1010	10)	SPR	n	Springe (unbedingt) $goto n;$
1011	11)	SP<	n	Springe bei negativem AC-Inhalt $if (ac < 0) \{goto n;\}$
1100	12)	SP=	n	Springe bei null $if (ac == 0) \{goto n;\}$
1101	13)	NOP	n	(no operation)
1110	14)	NOP	n	(no operation)
1111	15)	STOP	n	Beende das Maschinenprogramm $System.exit();$

wobei: ac Inhalt des Akkumulators, mem[n] Inhalt der n-ten Speicherstelle, n Adresse (0..63), Inhalt -512 .. 511