

2. Klausur 11/II (B)

Dauer: 3 Schulstunden (9:55 bis 12:15 Uhr)

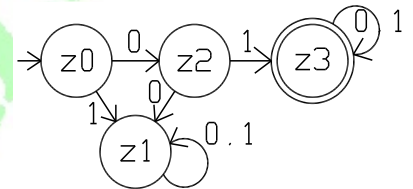
Name: www.r-krell.de

Hilfsmittel: normaler Taschenrechner

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

- ① Gegeben ist der erkennende Automat M durch $M = \{\mathcal{E}, \mathcal{Z}, \delta, Y, \{Z\}\}$, wobei $\mathcal{E} = \{x, y\}$ und $\mathcal{Z} = \{X, Y, Z\}$ ist. Die Übergangsfunktion $\delta: \mathcal{Z} \times \mathcal{E} \rightarrow \mathcal{Z}$ ist durch vollständige Aufzählung festgelegt: $\delta(X,a)=Z$, $\delta(X,b)=Y$, $\delta(Y,a)=Z$, $\delta(Y,b)=X$, $\delta(Z,a)=Z$ und $\delta(Z,b)=Z$.
- Zeichne den Automatengraph
 - Prüfe folgende Wörter und notiere, in welchem Zustand sie enden und ob sie vom Automaten akzeptiert werden oder nicht: (1) aab, (2) baab, (3) aba und (4) bbbab.
 - 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 mit "01" anfangen.



- Notiere die formale Beschreibung $A = \{\mathcal{E}, \mathcal{Z}, \delta, \dots\}$ des Automaten, wobei die Übergangsfunktion δ durch eine Automaten-tafel festgelegt werden soll.
 - Schreibe eine Java-Methode *beginntMit01*, die den Automaten schematisch nachbildet und genau dann *true* liefert, wenn das als String übergebene Wort mit "01" beginnt.
 - Schreibe eine kurze Java-Methode *startetMit01*, die den Befehl *indexOf* verwendet und genau dann *true* liefert, wenn das als String übergebene Wort mit "01" beginnt. (Hinweis: "abcdefg".*indexOf*("def") liefert 3, weil der Teilstring an der 3. Stelle von "abcdefg" beginnt, während "abc".*indexOf*("x") = -1 ist, weil "x" gar nicht in "abc" vorkommt.)
 - Konstruiere mit Wahrheitstafel (16 Zeilen) und KV-Diagramm eine Hardware-Schaltung, die bei vierstelligen Dualzahlen genau dann eine 1 ausgibt, wenn die vierstelligen Dualzahlen mit "01" beginnen.
 - Schreibe ein 1_AMOR-Maschinenprogramm, das genau dann eine 1 in Zelle 62 ausgibt, wenn die zehnstellige Dualzahl in Zelle 63 mit "01" beginnt. Sonst soll in Zelle 62 eine 0 bleiben. Beschreibe anfangs kurz die Idee und kommentiere dein Maschinenprogramm!
 - Bewerte: welcher der Tests auf den Start mit "01" wird (1) am schnellsten ausgeführt, (2) ist am schnellsten hergestellt bzw. (3) ist deiner Meinung nach am besten? (je mit kurzer Begründung)
- ③ Zeichne den Automatengraphen (oder gib statt dessen die Automaten-tafel an) für einen erkennenden Automaten, der beliebig lange Dualzahlen genau dann akzeptiert, wenn das erste und das letzte Zeichen verschieden sind. Auch das leere Wort, aber keine einstelligen Zahlen sollen akzeptiert werden (d.h. akzeptiert werden sollen beispielsweise "", "01", "01011" oder "10", während "1011" oder "0" nicht akzeptiert werden).
- ④ Ein 1_AMOR-Maschinenprogramm soll die Zahl in Zelle 63 auf folgende Weise verarbeiten und das Ergebnis wieder in Zelle 63 ausgeben: Zunächst wird die Zahl mit sich selbst addiert; dann wird die so entstandene Summe nochmals mit sich selbst addiert. (Beispiel: Wird anfangs die Zahl 9 eingegeben, so entsteht nach der ersten Addition das Zwischenergebnis $9+9=18$ und daraus das Endergebnis $18+18=36$).
- Der erste Befehl des benötigten Programms ist „0.) LAD 63“. Beschreibe alle 4 Schritte der Holphase für diesen Befehl sowie alle Schritte der Ausführungsphase!
 - Schreibe das ganze entsprechende 1_AMOR-Programm (kommentiert)

- c) Beschreibe in Deutsch, wie Endergebnis und Anfangszahl zusammen hängen.
 d) Nenne eine viel kürzere Möglichkeit (ein Maschinenprogramm mit 4 Befehlszeilen), um jeweils zum gleichen Endergebnis zu kommen. Welche Eigenschaft der Dualzahlen nutzt du aus?

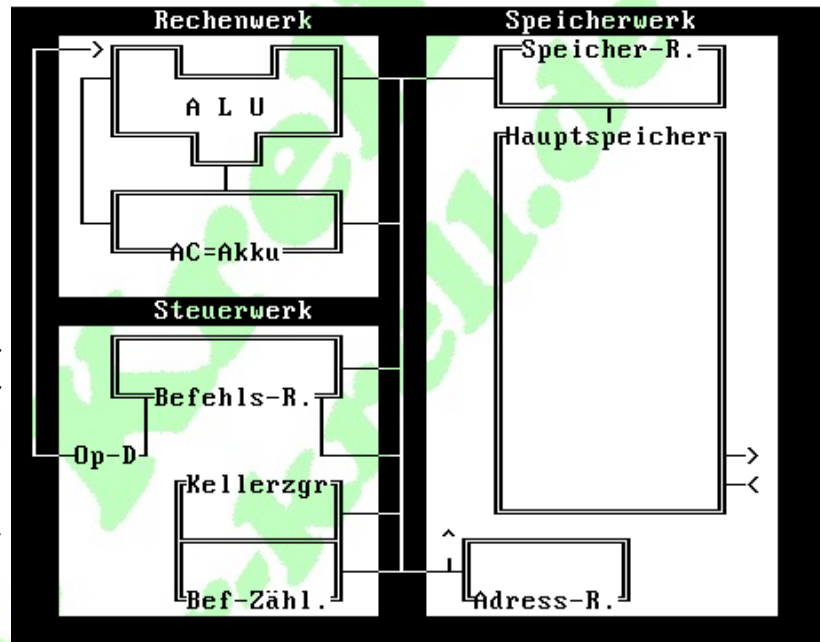
Modellrechner 1_AMOR -- Architektur und Befehlssatz

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 $ac = \sim ac; \quad // = -1-ac$
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