

Beispielklausur

# Einführung in die Informatik 1

Studiengang Künstliche Intelligenz

Wintersemester 2019/20

Termin	
Ort und Beginn	
Prüfungsdauer	90 Minuten
Hilfsmittel	ein handbeschriebenes Blatt DIN A 4
Hinweis	Ihre Darbietung muss den Lösungsweg und die eigenständige Bearbeitung erkennen lassen.

## Aufgabe 1 (Übung 1.4.a)

Wir betrachten TURING-Maschinen, die mit Zahlen in *unärer* Darstellung arbeiten. Im Unärsystem gibt es nur die einzige Ziffer 1. Natürliche Zahlen werden durch die ihrem Wert entsprechende Anzahl von Einsen dargestellt, bspw. wird die dezimale 5 unär durch 11111 dargestellt.

Entwickeln Sie eine Additionsmaschine, die zwei Unärzahlen addiert.

Beispielsweise sieht die Eingabe „4+2“ auf dem Band folgendermaßen aus:

			1	1	1	1	+	1	1			
--	--	--	---	---	---	---	---	---	---	--	--	--

**Aufgabe 2** (Übung 2.3.b, c und 3.1.b)

Geben Sie ein LOOP-Programm an, welches ...

- die Multiplikation von zwei Variablen  $x_1 * x_2$
- die Funktion  $f(n) = \begin{cases} 1 & \text{falls } n \text{ Quadratzahl ist} \\ 0 & \text{sonst} \end{cases}$  berechnet.
- Simulieren Sie (ohne Verwendung von LOOP) die Anweisung „IF  $x_i > x_j$  THEN  $P_1$  ELSE  $P_2$ “ durch ein WHILE-Programm.

**Aufgabe 3** (Übung 3.4)

Zeigen Sie, dass das Komplement einer entscheidbaren Menge entscheidbar ist.

**Aufgabe 4** (Übung 4.1)

Ordnen Sie folgenden Funktionen jeweils ihre Komplexitätsklasse (in O-Notation) zu und sortieren Sie die Komplexitätsklassen nach ihrer Größe.

$$a(n) = n^2 \cdot \lg(n) + 42$$

$$b(n) = 2^n + n^4$$

$$c(n) = 2^{2 \cdot n}$$

$$d(n) = 2^{n+3}$$

$$e(n) = \sqrt{n^5}$$

$$f(n) = 3 + 2 \cdot \sin(n)$$

$$g(n) = \begin{cases} 2 \cdot n & \text{falls } n \text{ ungerade} \\ \frac{n}{2} & \text{falls } n \text{ gerade} \end{cases}$$

**Aufgabe 5** (Übung 4.7)

Gegeben ist ein Rechner mit  $N = 8$  Stellen für Rechenoperationen, der im Zweierkomplement arbeitet.

- Ermitteln Sie die interne Darstellung der Zahl  $(-66)_{10}$ .
- Zeigen Sie, wie der Rechner die Subtraktion  $78_{10} - 66_{10}$  (auf die Addition zurückgeführt) durchführt.

### Aufgabe 6 (neue Übung)

Nachfolgend ist der Befehlssatz der Maschinensprache von BROOKSHEAR wiedergegeben:

Opcode	Operanden	Beschreibung
1	RXY	Lade Speicherwort aus Adresse XY in Register R
2	RXY	Lade Wert XY in Register R
3	RXY	Speichere Inhalt von Register R in Speicheradresse XY
4	0RS	Verschiebe Inhalt von Register R in Register S
5	RST	Addiere Inhalte der Register S und T und lege Ergebnis in R ab (alle Werte im Zweierkomplement)
6	RST	Addiere Inhalte der Register S und T und lege Ergebnis in R ab (alle Werte im an IEEE 754 angelehnten Gleitkommaformat 1+3+5 bit)
7	RST	OR-verknüpfe Inhalte der Register S und T und lege Ergebnis in R ab
8	RST	AND-verknüpfe Inhalte der Register S und T und lege Ergebnis in R ab
9	RST	XOR-verknüpfe Inhalte der Register S und T und lege Ergebnis in R ab
A	ROX	Rotiere den Inhalt von Register R um X Stellen nach rechts
B	RXY	Springe zum Befehl in Speicheradresse XY wenn der Inhalt von Register R gleich dem Inhalt von Register R0 ist
C	000	Halte die weitere Befehlsausführung an

Ein Emulator eines Rechners, der genau diese Maschinensprache verwendet, besitze folgenden Initialzustand:

### CPU

GPRs	PC	IR
0 00	00	0000
1 00		
2 00		
3 00		
4 00		
5 00		
6 00		
7 00		
8 00		
9 00		
A 00		
B 00		
C 00		
D 00		
E 00		
F 00		

**Decoded**

FETCH
DECODE
EXECUTE

Clear and Run

Clear CPU

Run

Step

### Memory

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	12	34	21	3F	82	21	A2	06	32	32	C0	00	00	00	00	00
1	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
2	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
3	00	00	00	00	44	00	00	00	00	00	00	00	00	00	00	00
4	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
5	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
6	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
7	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
8	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
9	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
A	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
B	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
D	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
E	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Welche Speicherstellen und welche Universalregister haben ihren Wert wie verändert, wenn der Startknopf geklickt wird und gewartet wird, bis der emulierte Rechner seinen HALT-Zustand erreicht hat.

Beschreiben Sie auch alle Zwischenzustände.

**Aufgabe 7** (Übung 6.4)

Entwerfen Sie ein Schaltnetz, das für vierstellige 8421-BCD-Codewörter ein Paritätsbit berechnet: Ist die Anzahl der Einsen im vierstelligen Codewort ungerade, dann soll das Paritätsbit Null sein. Ist die Anzahl der Einsen gerade, so soll es Eins sein („ungerade Parität“). Für irrelevante Codewörter ist der Wert des Paritätsbits egal.

BCD steht für *Binary Coded Decimal*, also binär codierte Dezimalziffer. Gültige 8421-BCD-Codewörter bestehen aus vier Bit und stellen jeweils eine dezimale Ziffer (0 bis 9) durch ein Halbbyte dar (0000 bis 1001).

- a) Erstellen Sie die Funktionstabelle für dieses Schaltnetz.
- b) Entwickeln Sie eine Funktionsgleichung in minimaler disjunktiver Normalform zur Berechnung des Paritätsbits.
- c) Zeichnen Sie unter Verwendung Ihrer Funktionsgleichung aus b) einen Logikschaltplan für das Schaltnetz zur Berechnung des Paritätsbits.

**Aufgabe 8** (neue Übung)

Gegeben ist ein direkter-abbildender Cache, der zu Beginn vollständig leer ist. Der Cache besitzt 256 Cache Lines. Jede Cache Line bietet Raum für 16 Bytes. Der Hauptspeicher ist byte-adressiert und unterstützt 16-bit-breite Adressen.

Es sollen Lesezugriffe auf die folgenden hexadezimalen Adressen durchgeführt werden:

0112, 0121, 0110, 1110, 0110, 0123, 012F und 1110.

Geben Sie an, bei welchem dieser Zugriffe ein Treffer verzeichnet wird.