Universität Augsburg
Institut für Informatik

Übung zur Vorlesung Informatik 1 WS 2020/21
Fakultät für Angewandte Informatik
Lehrprofessur für Informatik
Dr. Martin Frieb, Johannes Metzger 23.11.2020

Übungsblatt 4

Abgabe spätestens bis: 30.11.2020 10:00 Uhr

- Dieses Übungsblatt soll in den in der Übungsgruppe festgelegten Teams abgegeben werden (Einzelabgaben sind erlaubt, falls noch keine Teamzuteilung erfolgt ist).
- Die **Zeitangaben** geben zur Orientierung an, wie viel Zeit für eine Aufgabe später in der Klausur vorgesehen wäre; gehen Sie davon aus, dass Sie zum jetzigen Zeitpunkt wesentlich länger brauchen und die angegebene Zeit erst nach ausreichender Übung erreichen.

* leichte Aufgabe / ** mittelschwere Aufgabe / *** schwere Aufgabe

Aufgabe 13 * *(1K- und 2K-Codierung und -Dekodierung, 28 Minuten)*

Befüllen Sie die leeren Zellen in folgender Tabelle.
In der ersten Spalte steht jeweils eine Dezimalzahl x.
In den weiteren Spalten steht jeweils eine Codierung von x (die Spaltennamen geben jeweils die Codierung vor).
Liegt x außerhalb des Definitionsbereichs einer Codierung, so schreiben Sie ein Kreuz `X` in die Zelle.

x	$c_{1K,4}(x)$	$c_{2K,4}(x)$	$c_{1K,8}(x)$	$c_{2K,8}(x)$
3				
	1001			
			10010000	
-8				
		0111		
				01101001
128				

Geben Sie zu jedem Eintrag eine Rechnung oder Begründung an.

Aufgabe 14 *(Rechnen in 1K- und 2K-Codierungen, 20 Minuten)*

a) (*, Rechenoperationen auf Bitmustern, 12 Minuten)
Führen Sie folgende Rechenoperationen auf Bitmustern aus:

0101	$\oplus_{1K,4}$	0001	=	
0011	$\ominus_{1K,4}$	0111	=	
0101	$\oplus_{2K,4}$	0011	=	
1000	$\ominus_{2K,4}$	0101	=	
10010110	$\oplus_{1K,8}$	00110100	=	
01011010	$\ominus_{1K,8}$	00011111	=	
11010010	$\oplus_{2K,8}$	10110001	=	
10000000	$\ominus_{2K,8}$	00000001	=	

Geben Sie jeweils die zugehörige Rechnung an.

b) (*, Rechnen mit und ohne Überlauf, 8 Minuten)
Welchen Wert haben jeweils die folgenden Rechenausdrücke, wenn die Rechnung in der durch den Spaltennamen gegebenen Codierung durchgeführt wird?
Geben Sie das Ergebnis in Form einer Dezimalzahl an.

Rechenausdruck		$c_{1K,8}$	$c_{2K,8}$
$50 + 77$	=		
$41 + 97$	=		
$-17 - 110$	=		
$-95 - 33$	=		

Geben Sie jeweils die zugehörige Rechnung an.

Aufgabe 15 *(Formale Sprachen und Codierung, 15 Minuten)*

1. (*, 1 Minute)
   Welche Länge hat das Wort $0110101 \in \mathbb{B}^*$?

2. (*, 1 Minute)
   Geben Sie alle Elemente der Menge $\mathbb{B}^3$ an.

3. (*, 1 Minute)
   Wie viele Elemente hat die Menge $\mathbb{D}^4$?

4. (*, 1 Minute)
   Ordnen Sie die folgenden Worte über $\mathbb{B}$ bzgl. der lexikografischen Ordnung: $10, \epsilon, 01, 0001, 000, 001$.

5. (*, 1 Minute)
   Geben Sie alle Teilworte des Wortes $00111 \in \mathbb{B}^*$ an.

6. (*, 1 Minute)
   Wie viele Präfixe hat ein Wort $w \in \mathbb{A}^*$ mit der Länge $|w| = 20$?

7. (*, 1 Minute)
   Geben Sie möglichst formal die Menge aller Wörter in $\mathbb{B}^*$ an, die mit 00 enden.

8. (*, 1 Minute)
   Geben Sie möglichst formal die Menge aller Wörter in $\mathbb{D}^*$ an, die mindestens die Länge 2 haben.

9. (*, 1 Minute)
   Bestimmen Sie alle Elemente der Menge $\mathbb{A}^* \setminus \mathbb{A}^+$.

10. (*, 1 Minute)
    Geben Sie möglichst formal die Menge aller nicht-leeren Wörter in $\mathbb{B}^*$ an, die nur aus 0en bestehen.

11. (**, 3 Minuten)
    Für ein beliebiges Alphabet A, $n \in \mathbb{N}_0$ und $w \in A^*$ sei $w^n \in A^*$ wie folgt induktiv nach n definiert:

    - Induktionsanfang $n = 0$: $w^0 := \epsilon$.
    - Induktionsschritt $n \to n + 1$: $w^{n+1} := ww^n$ für $n \geq 1$.

    Schreiben Sie folgende Wörter aus $\mathbb{B}^*$ aus (ohne Verwendung hochgestellter Zahlen):
    0^5, $(100)^2$, 100^2.

12. (**, 1 Minute)
    Betrachten Sie die Codierung $c_{Z1} : \mathbb{D} \to \mathbb{B}^*$, $c_{Z1}(d) := 0^{d+1}$. Geben Sie alle Wörter $w \in \mathbb{D}^*$ an, für die $c_{Z1}(w) = 0^4$ gilt (vgl. *Codierung von Wörtern*, Skript, Seite 8).

13. (*, 1 Minute)
    Betrachten Sie die Codierung $c_{Z2} : \mathbb{D} \to \mathbb{B}^*$, $c_{Z2}(d) := 0^{d+1}1$. Codieren Sie folgende Wörter: 0511, 110 (vgl. *Codierung von Wörtern*, Skript, Seite 8).

Aufgabe 16 *(Funktionen, Felder und bitweise Operatoren, 26 Minuten)*

Erstellen Sie ein kompilierbares und ausführbares C-Programm mit einer `main`-Funktion und weiteren Funktionen nach folgenden Vorgaben.

- (**, 3 Minuten)
  Legen Sie eine systemunabhängige symbolische Konstante `INT_BIT` für die Anzahl der Bits des Datentyps `int` an. Tipps:
  - Der Datentyp `int` ist plattformabhängig unterschiedlich groß. Verwenden Sie den Operator `sizeof`, um die Anzahl der Byte zu ermitteln, die ein `int` benötigt.
  - Symbolische Konstanten haben Sie bereits im Vorkurs verwendet (z.B. `INT_MAX`). Sie können selbst derartige Konstanten anlegen, indem Sie `#define INT_BIT 12345` schreiben. Setzen Sie für die Zahl eine passende Berechnung der Anzahl der Bits ein.
- (*, 3 Minuten)
  Legen Sie eine Funktion `void print_bits(int b[], int size)` an, die nacheinander (ohne Trennzeichen) in umgekehrter Reihenfolge die ersten `size` Komponenten des Feldes `b` auf Kommandozeile ausgibt (also die Komponente `b[0]` zuletzt).
- (*, 2 Minuten)
  Legen Sie eine Funktion `void init_bits(int b[], int size)` an, die die ersten `size` Komponenten des Feldes `b` mit dem Wert `0` initialisiert.
- (***, 5 Minuten)
  Legen Sie eine Funktion `void get_bits(int b[], int n)` an, die die Bits der Codierung der Zahl `n` auf Ihrem System im Feld `b` speichert (also das Bit b_i in der Komponente `b[i]` speichert für die Bits $b_0, \ldots, b_{INT\_BIT-1}$). Verwenden Sie dabei zur Berechnung der Bits ausschließlich bitweise Operatoren.
- (***, 8 Minuten)
  Legen Sie eine Funktion `int get_int(int b[])` an, die die Komponenten des Feldes `b` als 2K-Codierung in `INT_BIT` Bits einer ganzen Zahl interpretiert, diese Zahl berechnet und zurückgibt. Verwenden Sie dabei zur Berechnung der ganzen Zahl ausschließlich bitweise Operatoren.
- (*, 5 Minuten)
  Deklarieren Sie in der `main`-Funktion ein `int`-Feld `b` mit `INT_BIT` Komponenten und testen Sie mit diesem Feld die obigen Funktionen.

Vergessen Sie nicht, die notwendigen Bibliotheken einzubinden.