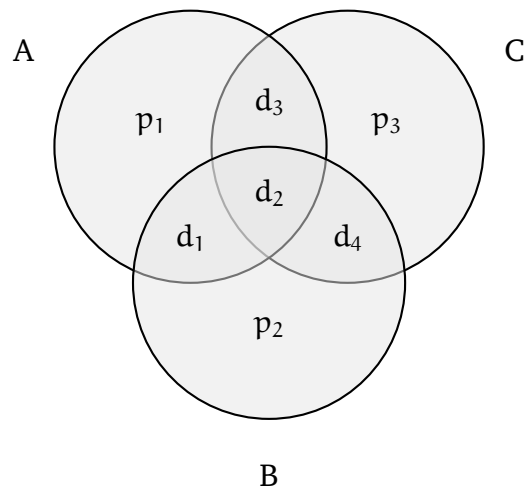


## Rechnerarchitektur im Sommersemester 2019 Übungsblatt 12

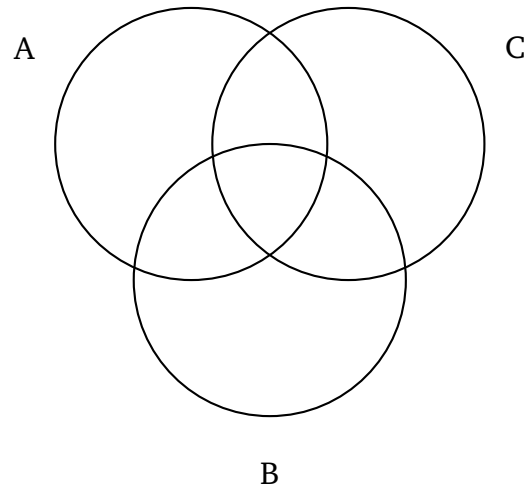
### Aufgabe 61: (H) Fehlererkennungscode

(8 Pkt.)

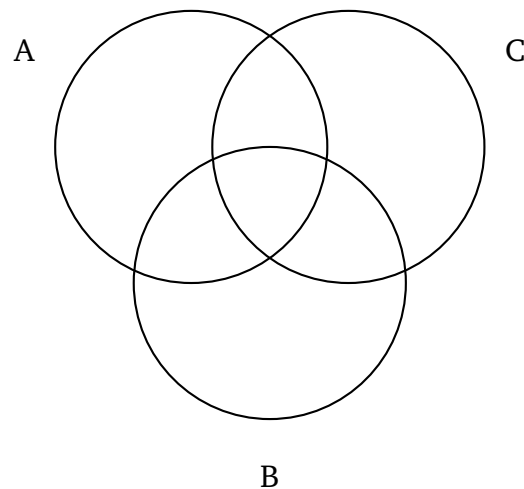
Wir gehen von folgender Struktur der Code-Wörter  $d_1 d_2 d_3 d_4 p_1 p_2 p_3$  aus. Wobei  $d_i (i \in \{1, 2, 3, 4\})$  für das jeweilige Datenbit und  $p_j (j \in \{1, 2, 3\})$  für das jeweilige Prüf- bzw. Paritätsbit steht. Die Paritätsbits zur Fehlererkennung bzw. Fehlerkorrektur für ein Datenwort  $d_1 d_2 d_3 d_4$  können anschaulich mit Hilfe eines Venn-Diagramms berechnet werden, in welchem sich die Bits wie folgt anordnen:



- a. Berechnen Sie unter Verwendung des folgenden Venn-Diagramms die Prüfbits für das Datenwort **1111**. Verwenden Sie dazu **gerade Parität**. Tragen Sie zunächst die Datenbits in die für die Berechnung sinnvollen (Schnitt-)Mengen ein.

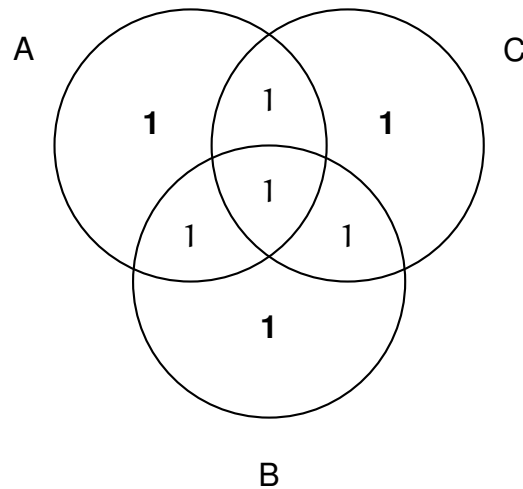


- b. Gehen Sie nun davon aus, dass Sie ein mit dem zuvor beschriebenen Code codiertes Code-Wort **0001010** empfangen haben. Es wurde **gerade Parität** verwendet. Handelt es sich um ein gültiges Codewort? Falls nein, treffen Sie eine Aussage darüber, an welcher/welchen Stelle/-Stellen mutmaßlich (ein) Bitfehler aufgetreten ist/sind. Verwenden Sie zur Berechnung das folgende Venn-Diagramm. Korrigieren Sie (falls möglich/nötig) den/die Fehler **innerhalb** des Venn-Diagramms und geben Sie das (ggf. korrigierte) 4-Bit Datenwort an.

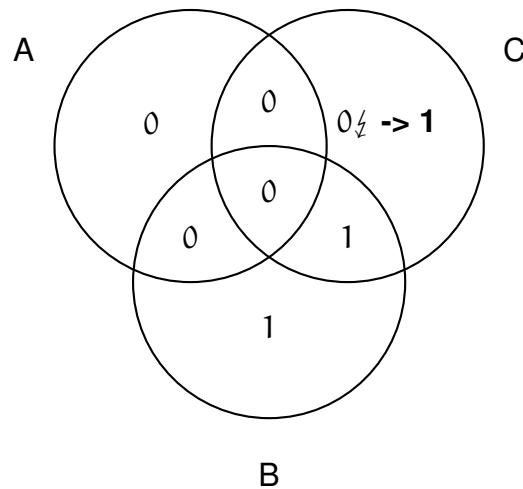


**Antwort:**

- a. Damit die Anzahl der mit 1 belegten Bits in den Mengen A, B und C gerade ist, müssen alle 3 Paritätsbit  $p_1$ ,  $p_2$  und  $p_3$  auf 1 gesetzt werden (sie zählen mit zur Anzahl bzw. prüfen sich selbst).



- b.



Es ist kein gültiges Codewort. Bis auf Paritätsbit  $p_3$  passt die Parität über alle Mengen. Dementsprechend ist davon auszugehen, dass Paritätsbit  $p_3$  gekippt ist und 1 sein müsste, zumindest wenn man von einer minimalen Anzahl an gekippter Bits ausgeht. Mehr als 1 Bit kann ohnehin nicht mit Sicherheit korrigiert werden. Das Datenwort ist nicht betroffen und lautet 0001.

## Aufgabe 64: (H) Einfachauswahlaufgabe: Darstellung von Speicherinhalten

(5 Pkt.)

Für jede der folgenden Fragen ist eine korrekte Antwort auszuwählen („1 aus n“).

a) Wie viele Bit stehen im ursprünglichen ASCII-Code zur Kodierung eines Zeichens zur Verfügung?			
(i) 1	(ii) 7	(iii) 16	(iv) 128
b) Die Dezimalzahl 16.909.060 (01020304 Hexadezimal) soll als 32-Bit-Integer-Wert (Wortbreite) ab Speicheradresse 0000 gespeichert werden. Dabei kommt die Little Endian Byte-Anordnung zum Einsatz. Welche Antwort entspricht der resultierenden Speicherbelegung?			
(i)	(ii)	(iii)	(iv)
Adresse	Adresse	Adresse	Adresse
Wert	Wert	Wert	Wert
0000 01	0000 02	0000 04	0000 03
0001 02	0001 01	0001 03	0001 01
0002 03	0002 04	0002 02	0002 02
0003 04	0003 03	0003 01	0003 04
c) Welche Operation kann auf zwei gleichlange Codewörter angewendet werden, um durch Zählen der 1en im Ergebnis den Hamming-Abstand der Codewörter zu bestimmen?			
(i) AND	(ii) OR	(iii) XOR	(iv) NOR
d) Gehen Sie nun davon aus, dass Sie folgendes Code-Wort <b>1010001</b> empfangen haben. Es wurde <b>gerade Parität</b> verwendet. Bei der Übertragung ist ein einzelner Bitfehler aufgetreten. Welches Paritätsbit ist betroffen?			
(i) A	(ii) B	(iii) C	(iv) keins
e) Angenommen ein Speicherwort wird in einem kurzen Zeitintervall $k$ mal gelesen oder geschrieben und befindet sich nach dem ersten Zugriff im Cache. Wie berechnet sich die Trefferrate (Hit Ratio) $h$ ?			
(i) $h = \frac{k-1}{k}$	(ii) $h = \frac{k}{k-1}$	(iii) $h = (k-1) \cdot (k)$	(iv) $h = \frac{k}{k}$

**Antwort:**

- a. (ii)
- b. (iii)
- c. (iii)
- d. (ii)
- e. (i)