

Schaltelemente

Boolesche Algebra

KV-Diagramm

Aussagenlogik

Logelei

Aussagenlogik Übersetzungen

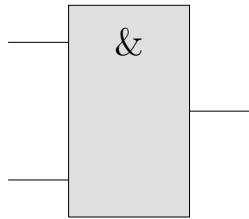
Logelei Python

Programm in PureBasic

Logelei

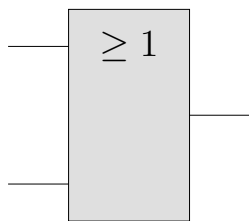
↑ Schaltelemente

Und Konjunktion a und b



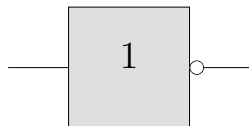
a	b	$a \wedge b$
0	0	0
0	1	0
1	0	0
1	1	1

Oder Disjunktion a oder b



a	b	$a \vee b$
0	0	0
0	1	1
1	0	1
1	1	1

Negation nicht a



a	\bar{a}
0	1
1	0

Mit diesen elementaren Bauteilen lassen sich alle Computerschaltungen realisieren.

↑ Gesetze der Booleschen Algebra

Kommutativgesetz:	$a \vee b = b \vee a$	$a \wedge b = b \wedge a$
Assoziativgesetz:	$(a \vee b) \vee c = a \vee (b \vee c)$	$(a \wedge b) \wedge c = a \wedge (b \wedge c)$
Absorptionsgesetz:	$a \vee (a \wedge b) = a$	$a \wedge (a \vee b) = a$
Distributivgesetz:	$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$	$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$
Idempotenzgesetz:	$a \vee \bar{a} = 1$	$a \wedge \bar{a} = 0$
	$a \vee a = a$	$a \wedge a = a$
de Morgan-Gesetze:	$\overline{a \vee b} = \bar{a} \wedge \bar{b}$	$\overline{a \wedge b} = \bar{a} \vee \bar{b}$
neutrale Elemente:	$a \vee 0 = a$	$a \wedge 1 = a$
doppelte Negation:	$\overline{\bar{a}} = a$	

1. Zeige mit den Gesetzen der Booleschen Algebra die Gleichheit der Terme.

- | | |
|---|---|
| a) $(a \wedge \bar{b}) \vee (a \wedge \bar{b} \wedge c) = a \wedge \bar{b}$ | b) $(a \vee b) \wedge (\bar{a} \vee b) \wedge (a \vee \bar{b}) \wedge (\bar{a} \vee \bar{b}) = 0$ |
| c) $(\overline{a \wedge \bar{b} \vee c}) \wedge (\bar{a} \vee b \vee \bar{c}) = \bar{a} \vee \bar{c}$ | d) $\overline{\overline{a \wedge \bar{b} \vee c}} \vee (a \wedge c) = a \wedge (b \vee c)$ |

2. Gib für die booleschen Funktionen möglichst einfache Terme an.

			a)	b)	c)	d)	e)	f)
<i>a</i>	<i>b</i>	<i>c</i>	<i>t</i>	<i>u</i>	<i>v</i>	<i>x</i>	<i>y</i>	<i>z</i>
0	0	0	1	0	1	1	?	0
0	0	1	0	0	1	1	0	?
0	1	0	1	1	0	0	1	?
0	1	1	0	1	0	?	0	0
1	0	0	0	0	1	0	1	0
1	0	1	1	1	1	?	0	0
1	1	0	1	0	1	0	0	1
1	1	1	1	1	0	1	?	1

↑ Boolesche Algebra

Gib für die booleschen Funktionen möglichst einfache Terme an.

			a)	b)	c)	d)	e)	f)
<i>a</i>	<i>b</i>	<i>c</i>	<i>t</i>	<i>u</i>	<i>v</i>	<i>x</i>	<i>y</i>	<i>z</i>
0	0	0	1	0	1	1	1	0
0	0	1	0	0	1	1	0	0
0	1	0	1	1	0	0	1	0
0	1	1	0	1	0	0	0	0
1	0	0	0	0	1	0	1	0
1	0	1	1	1	1	1	0	0
1	1	0	1	0	1	0	0	1
1	1	1	1	1	0	1	0	1

a) $(\bar{a} \wedge \bar{c}) \vee (a \wedge c) \vee (b \wedge \bar{c})$

b) $(\bar{a} \wedge b) \vee (a \wedge c)$

c) $\bar{b} \vee (a \wedge \bar{c})$

d) $(\bar{a} \wedge \bar{b}) \vee (a \wedge c)$

e) $(\bar{a} \wedge \bar{c}) \vee (\bar{b} \wedge \bar{c}) = (\bar{a} \vee \bar{b}) \wedge \bar{c}$

f) $a \wedge b$

↑ Boolesche Algebra

a) PureBasic

```
OpenConsole()
```

```
For a = #False To #True
```

```
    For b = #False To #True
```

```
        For c = #False To #True
```

```
            PrintN(Str(a)+Str(b)+Str(c)+ " " + Str( ((Not c) Or a) And ( (Not a) Or b Or c ) ) )
```

```
        Next c
```

```
    Next b
```

```
Next a
```

```
Input()
```

b) Python

```
for A in [False, True]:
```

```
    for B in [False, True]:
```

```
        for C in [False, True]:
```

```
            D1 = ...
```

```
            D2 = ...
```

```
            D3 = ...
```

```
            print(int(A), int(B), int(C), " ", int(D1 and D2 and D3))
```

↑ Boolesche Algebra

Sei $\mathbb{B} = \{0, 1\}$ die Boolesche Menge, auf der folgende Verknüpfungen erklärt sind (x, y Boolesche Variablen):

Addition

$$x + y = x \vee y \quad (= \text{Max}(x, y)) \quad \text{Beachte den Unterschied } 1 + 1 = 1 \text{ zur Addition von Dualzahlen.}$$

Multiplikation

$$x \cdot y = x \wedge y \quad (= \text{Min}(x, y))$$

Das vereinfacht die Schreibweise Boolescher Terme, statt $(a \wedge \bar{b}) \vee (a \wedge \bar{b} \wedge c)$ ist $a\bar{b} + a\bar{b}c$ möglich.

Gib für die Booleschen Funktionen X, Y und Z einen Term an.

a	b	c	X	Y	Z
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	1	1	1
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	0	1	1
1	1	0	0	0	1
1	1	1	0	0	0

$$X = \bar{a}b\bar{c}$$

$$Y = \bar{a}b\bar{c} + a\bar{b}c$$

$$Z = \bar{a}b\bar{c} + a\bar{b}c + ab\bar{c} \quad \text{disjunktive Normalform}$$

Es sollte nun klar sein, dass zu jeder Booleschen Funktion ein zugehöriger Boolescher Term in disjunktiver Normalform ermittelt werden kann.

Wir wollen der Frage nachgehen, wie dieser Term vereinfacht werden kann.

Betrachten wir die Boolesche Funktion:

a	b	X
0	0	1
0	1	0
1	0	0
1	1	1

Die Funktion $X = \bar{a}\bar{b} + ab$ kann mit einem Diagramm veranschaulicht werden.

	\bar{a}	a
\bar{b}	00	10
b	01	11

a wird für $\{(1,0), (1,1)\}$ 1,

b wird für $\{(0,1), (1,1)\}$ 1,

ab wird nur auf dem gemeinsamen Element $(1,1)$ 1.

Die Randbeschriftungen markieren also die Bereiche, für die die elementaren Terme jeweils 1 werden.

Wie lautet der Boolesche Term?

a)

	\bar{a}	a
\bar{b}		
b		

b)

	\bar{a}	a
\bar{b}		
b		

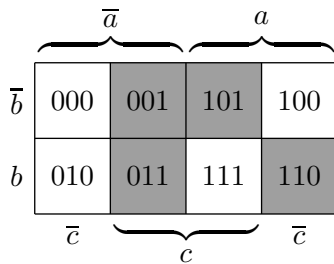
c) Wie sieht das KV-Diagramm für den Term $a + b$ aus?

↑ KV-Diagramm

Betrachten wir die Boolesche Funktion:

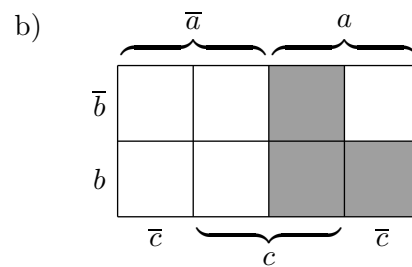
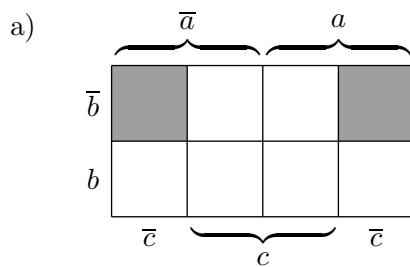
a	b	c	X
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Das Karnaugh-Veitch-Diagramm muss aus 8 Feldern bestehen.



Die Unterteilung ist so angelegt, dass sich benachbarte Felder lediglich um eine Stelle unterscheiden.

Wie lautet der Boolesche Term?



↑ Idee des KV-Diagramms

a	b	c	X
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

\bar{b}	c	$\bar{b}c$	$ab\bar{c}$	$\bar{b}c + ab\bar{c}$
1	0	0	0	0
1	1	1	0	1
0	0	0	0	0
0	1	0	0	0
1	0	0	0	0
1	1	1	0	1
0	0	0	1	1
0	1	0	0	0

	\bar{a}		a	
\bar{b}	000	001	101	100
b	010	011	111	110
	\bar{c}		c	

$$X = \bar{b}c + ab\bar{c}$$

Jede Boolesche Funktion (mit 3 Variablen) und damit jeder Term legt eine Teilmenge der Menge

$$\mathbb{A} = \{(0, 0, 0), (0, 0, 1), (0, 1, 0), (0, 1, 1), (1, 0, 0), (1, 0, 1), (1, 1, 0), (1, 1, 1)\} \quad \text{fest, hier ist es}$$

$$\mathbb{B} = \{(0, 0, 1), (1, 0, 1), (1, 1, 0)\}$$

Auf dieser Teilmenge ist die Funktion (der Term) 1, sonst 0.

Der Term \bar{b} wird auf $\mathbb{C} = \{(0, 0, 0), (0, 0, 1), (1, 0, 1), (1, 0, 0)\}$ 1,

der Term c auf $\mathbb{D} = \{(0, 0, 1), (0, 1, 1), (1, 0, 1), (1, 1, 1)\}$.

$\bar{b}c$ wird auf der Schnittmenge $\mathbb{C} \cap \mathbb{D} = \{(0, 0, 1), (1, 0, 1)\}$ 1.

$ab\bar{c}$ wird auf der $\mathbb{E} = \{(1, 1, 0)\}$ 1, das ist die Schnittmenge der zu a , b und \bar{c} gehörenden Teilmengen.

Die Funktion X nimmt damit auf der Menge $(\mathbb{C} \cap \mathbb{D}) \cup \mathbb{E}$ den Wert 1 an,

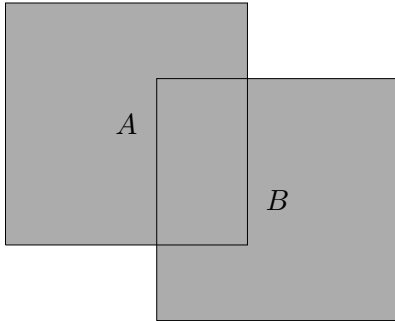
der Funktionsterm lautet: $\bar{b}c + ab\bar{c}$

+ bedeutet im KV-Diagramm die Vereinigung von Feldern,

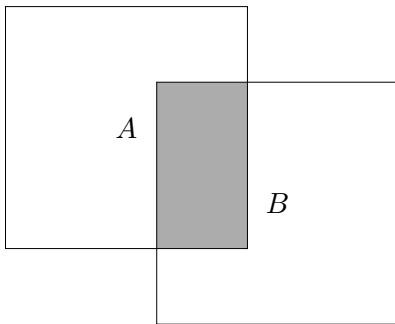
· die Bildung der Schnittmenge.

↑ Mengenoperationen

$A \cup B$ Vereinigung

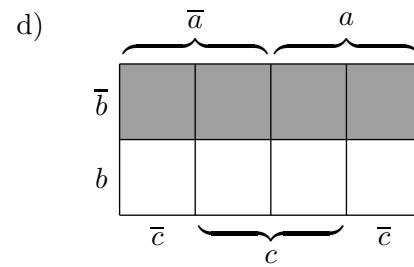
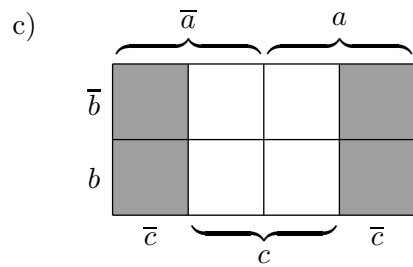
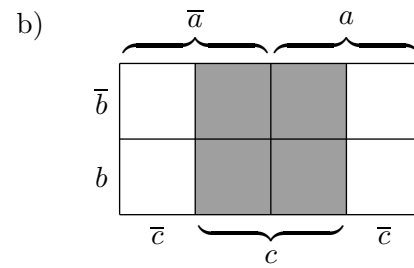
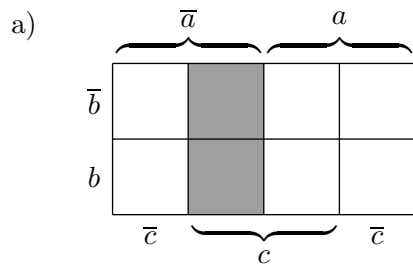


$A \cap B$ Schnitt



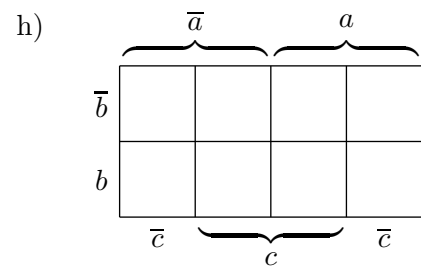
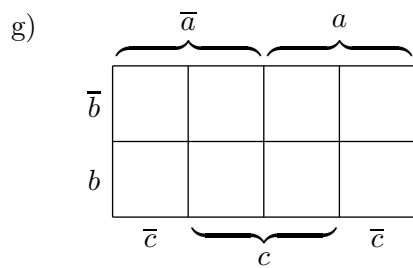
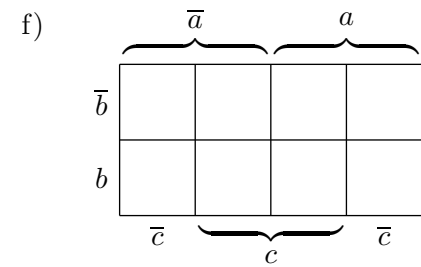
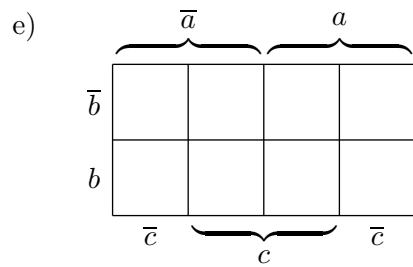
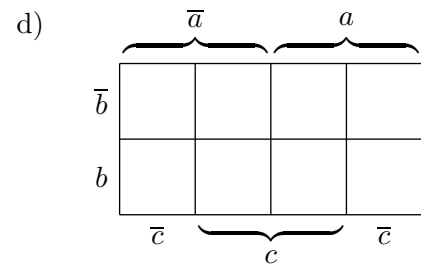
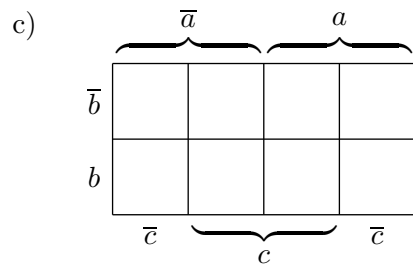
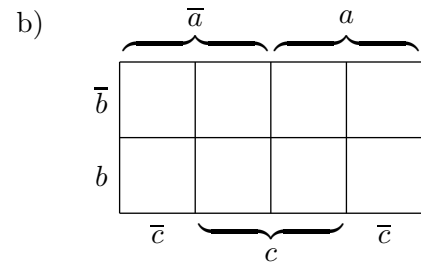
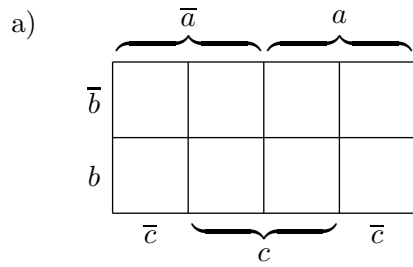
↑ KV-Diagramm

Wie lautet der Boolesche Term?



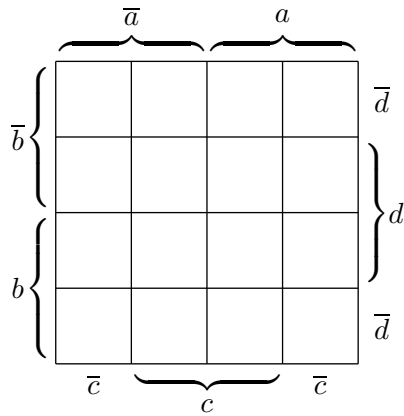
e) Wie sieht das KV-Diagramm für den Term $(\bar{a} + c)b + \bar{c}b$ aus?

↑ KV-Diagramm Vorlagen



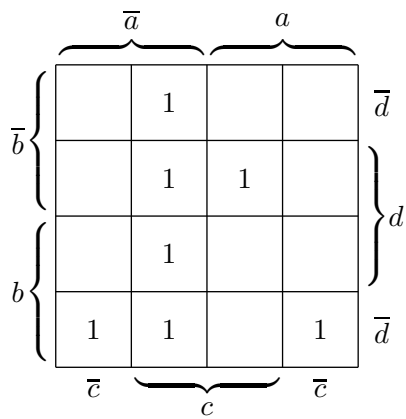
↑ KV-Diagramm

Das Karnaugh-Veitch-Diagramm für 4 Variablen muss aus 16 Feldern bestehen.



Wie sieht das KV-Diagramm für den Term $(\bar{a} + b)(c + \bar{d})$ aus?

↑ KV-Diagramm



Gib einen minimalen Term an.

↑

↑ KV-Diagramm

$$\bar{a}c + \bar{b}cd + b\bar{c}\bar{d}$$

↑ Aussagenlogik

Eine Aussage ist entweder wahr oder falsch.
Wir bezeichnen Aussagen mit großen Buchstaben.
Aussagen können verknüpft werden mit

und	$A \wedge B$
oder	$A \vee B$
wenn ..., dann ...	$A \rightarrow B$
... genau dann, wenn ...	$A \leftrightarrow B$

A	B	$A \rightarrow B$
0	0	1
0	1	1
1	0	0
1	1	1

$A \rightarrow B$ mache man sich am Beispiel klar:
Wenn Frau L. Musik hört, benutzt sie einen Kopfhörer.
Diese Aussage ist nur falsch, wenn ...

A	B	$A \leftrightarrow B$
0	0	1
0	1	0
1	0	0
1	1	1

Für Umformungen kann $A \rightarrow B = \overline{A} \vee B$ nützlich sein.
 $\overline{A} \vee B$ ist nur 0, wenn $A = 1$ und $B = 0$ vorliegt.

Die Gesetze der Booleschen Algebra können neu formuliert werden.
Hierzu wird das Gleichheitszeichen durch das Äquivalenzzeichen ersetzt, z.B. $\overline{A \vee B} \leftrightarrow \overline{A} \wedge \overline{B}$.

Eine Aussage, die stets wahr ist, heißt Tautologie, in Zeichen: \Rightarrow (Implikation), \Leftrightarrow (Äquivalenz).
Wir schreiben also genauer:

$$\overline{A \vee B} \Leftrightarrow \overline{A} \wedge \overline{B} \quad \text{gemeint: } \overline{A \vee B} \Leftrightarrow (\overline{A} \wedge \overline{B}), \text{ also stets } (\dots) \Leftrightarrow (\dots)$$

$$(\dots) \Rightarrow (\dots)$$

Weiteres

$$(A \rightarrow B) \wedge (B \rightarrow C) \Rightarrow (A \rightarrow C)$$

$$A \wedge (A \rightarrow B) \Rightarrow B$$

$$A \rightarrow B \Leftrightarrow \overline{B} \rightarrow \overline{A} \quad \text{Kontraposition}$$

$$A \leftrightarrow B \Leftrightarrow (A \rightarrow B) \wedge (B \rightarrow A)$$

$$A \leftrightarrow B \Leftrightarrow (A \rightarrow B) \wedge (\overline{A} \rightarrow \overline{B})$$

Überprüfe

$$\overline{B} \wedge (A \rightarrow B) \Rightarrow \overline{A}$$

$$(A \rightarrow B) \wedge (\overline{A} \rightarrow B) \Leftrightarrow B$$

$$\overline{A \leftrightarrow B} \Leftrightarrow A \leftrightarrow \overline{B}$$

↑

↑ Logelei

Die SchülerInnen sind dann und nur dann glücklich, wenn kein Test geschrieben wird. Wenn die SchülerInnen glücklich sind, fühlt sich der Lehrer wohl. Aber wenn sich der Lehrer wohl fühlt, hat er keine Lust, zu unterrichten. Und wenn er keine Lust hat, zu unterrichten, wird ein Test geschrieben. Sind die Schüler glücklich?

↑ Logelei

Die SchülerInnen sind dann und nur dann glücklich, wenn kein Test geschrieben wird. Wenn die SchülerInnen glücklich sind, fühlt sich der Lehrer wohl. Aber wenn sich der Lehrer wohl fühlt, hat er keine Lust, zu unterrichten. Und wenn er keine Lust hat, zu unterrichten, wird ein Test geschrieben. Sind die Schüler glücklich?

- S : Die SchülerInnen sind glücklich.
 T : Es wird ein Test geschrieben.
 L : Der Lehrer fühlt sich wohl.
 U : Der Lehrer hat Lust zu unterrichten.

$$\begin{array}{l} S \leftrightarrow \neg T \\ S \rightarrow L \\ L \rightarrow \neg U \\ \neg U \rightarrow T \\ \hline \neg S \end{array} \quad \text{Negation } \neg \quad \quad \quad S \vee \neg S, \quad S \rightarrow \dots \neg S$$

↑ Aussagenlogik Übersetzungen

Übersetze die Sätze in die Sprache der Aussagenlogik.

- a) Die Erde kreist um die Sonne, aber der Papst glaubt das nicht.
- b) Klaus kommt genau dann zur Party, wenn Anna nicht kommt oder wenn Olaf die Drinks mixt.
- c) Wenn Sofie einen Witz macht, dann lachen Hans und Erna, es sei denn, dass Klaus oder Peter die Stimmung verdirbt.

↑ Aussagenlogik Übersetzungen

Übersetze die Sätze in die Sprache der Aussagenlogik.

- a) Die Erde kreist um die Sonne, aber der Papst glaubt das nicht.

E : Die Erde kreist um die Sonne.

P : Der Papst glaubt, dass die Erde um die Sonne kreist.

$$E \wedge \neg P$$

- b) Klaus kommt genau dann zur Party, wenn Anna nicht kommt oder wenn Olaf die Drinks mixt.

K : Klaus kommt zur Party.

A : Anna kommt zur Party.

O : Olaf mixt die Drinks.

$$K \leftrightarrow \neg A \vee O$$

- c) Wenn Sofie einen Witz macht, dann lachen Hans und Erna, es sei denn, dass Klaus oder Peter die Stimmung verdirbt.

S : Sofie macht einen Witz.

H : Hans lacht.

E : Erna lacht.

K : Klaus verdirbt die Stimmung.

P : Peter verdirbt die Stimmung.

$$S \rightarrow (H \wedge E \leftrightarrow \neg K \wedge \neg P)$$

↑ Logelei

Anna sagt: „Bettina lügt“.

Bettina sagt: „Claudia lügt“.

Claudia sagt: „Anne und Bettina lügen“.

Wer lügt denn nun?

↑ Logelei

Anna sagt: „Bettina lügt“.
Bettina sagt: „Claudia lügt“.
Claudia sagt: „Anne und Bettina lügen“.
Wer lügt denn nun?

A : Anna sagt die Wahrheit.
 B : Bettina sagt die Wahrheit.
 C : Claudia sagt die Wahrheit.

$A \leftrightarrow \neg B$
 $B \leftrightarrow \neg C$
 $C \leftrightarrow \neg A \wedge \neg B$

Wir suchen eine Variablenbelegung, so dass die Aussagen wahr sind (Erfüllbarkeits-Problem).

```
def äqui(X,Y):
    return X == Y          # alternativ folgt(X,Y) and folgt(Y,X)

for A in [False,True]:
    for B in [False,True]:
        for C in [False,True]:
            D1 = äqui(A, not B)
            D2 = äqui(B, not C)
            D3 = äqui(C, not A and not B)
            print(int(A), int(B), int(C), " ", int(D1 and D2 and D3))
```

Bettina sagt die Wahrheit.
 $B = 1, A = 0, C = 0$

↑ Programm in PureBasic

```
OpenConsole ()

For n=0 To 1
  For m=0 To 1
    For k=0 To 1

      A=n: B=m: C=k
      D1= ((Not A) | (Not B)) & (B|A)
      D2= ((Not B) | (Not C)) & (C|B)
      D3= ((Not C) | (Not (A|B))) & (A|B|C)

      If D1 & D2 & D3
        PrintN("ABC "+ Str(A)+Str(B)+Str(C))
      EndIf

    Next k
  Next m
Next n
Input ()
```

Etwas eleganter:

```
OpenConsole ()

For n=0 To 8
  v=n
  A=v % 2: v/2
  B=v % 2: v/2
  C=v % 2: v/2

  D1= ((Not A) | (Not B)) & (B|A)
  D2= ((Not B) | (Not C)) & (C|B)
  D3= ((Not C) | (Not (A|B))) & (A|B|C)

  If D1 & D2 & D3
    PrintN("ABC "+Str(A)+Str(B)+Str(C))
  EndIf
Next n
Input ()
```

↑ Logelei

Ein Mann schreit: „Wer von Euch Halunken hat den Ball in mein Fenster geworfen?“

Zitternd stehen vier Kinder da.

Anne sagt: „Emil war es.“

Emil sagt: „Gustav war es.“

Fritz sagt: „Ich war es nicht.“

Gustav sagt: „Emil lügt.“

Ein Passant sagt: „Eines der Kinder war es, aber Vorsicht: Nur eines der Kinder sagt die Wahrheit.“

Frage: Wer hat den Ball geworfen?

↑ Logelei

Ein Mann schreit: „Wer von Euch Halunken hat den Ball in mein Fenster geworfen?“

Zitternd stehen vier Kinder da.

Anne sagt: „Emil war es.“

Emil sagt: „Gustav war es.“

Fritz sagt: „Ich war es nicht.“

Gustav sagt: „Emil lügt.“

Ein Passant sagt: „Eines der Kinder war es, aber Vorsicht: Nur eines der Kinder sagt die Wahrheit.“

Frage: Wer hat den Ball geworfen?

Gustav sagt die Wahrheit. Fritz war es.

↑ Besuch der Meiers

Herr und Frau Meier haben die drei Söhne Tim, Kay und Uwe.

Wenn Herr Meier kommt, dann bringt er auch seine Frau mit.

Mindestens einer der beiden Söhne Uwe und Kay kommt.

Entweder kommt Frau Meier oder Tim.

Entweder kommen Tim und Kay oder beide nicht.

Und wenn Uwe kommt, dann auch Kay und Herr Meier.

↑ Besuch der Meiers

Herr und Frau Meier haben die drei Söhne Tim, Kay und Uwe.

Wenn Herr Meier (H) kommt, dann bringt er auch seine Frau (F) mit.

Mindestens einer der beiden Söhne Uwe (U) und Kay (K) kommt.

Entweder kommt Frau Meier oder Tim (T).

Entweder kommen Tim und Kay oder beide nicht.

Und wenn Uwe kommt, dann auch Kay und Herr Meier.

$$\begin{array}{ll} H \rightarrow F & (\neg H \vee F) \\ U \vee K & \\ \neg(F \leftrightarrow T) & (F \wedge \neg T) \vee (\neg F \wedge T) \\ T \leftrightarrow K & \\ U \rightarrow K \wedge H & \end{array}$$

Wir suchen eine Variablenbelegung, so dass die Aussagen wahr sind (Erfüllbarkeits-Problem).

```
def folgt(X,Y):
```

```
    return not X or Y
```

```
def äqui(X,Y):
```

```
    return X == Y      # alternativ folgt(X,Y) and folgt(Y,X)
```

Tim und Kay kommen zu Besuch.

↑ Wahl des Studienfachs

Die Drillinge Anton, Bernd und Christian stehen davor, sich an der Universität entweder für das Studienfach Mathematik oder für Informatik einzuschreiben, und diskutieren darüber, wer sich für welches Fach entscheiden sollte. Dabei hat jeder der Drei bestimmte Wunschvorstellungen. So verkündet Bernd sofort, dass er Mathematik studieren will, falls einer seiner beiden Brüder sich für Informatik einschreiben sollte. Christian hingegen will Informatik studieren, wenn Bernd dies ebenfalls tut oder wenn Anton sich für Mathematik entscheidet. Um die Angelegenheit noch komplizierter zu machen, mischt sich Anton ein und sagt: „Wenn Bernd Mathe studiert, dann tu ich das auch!“ Daraufhin erwidert Christian: „Solltest du dich aber doch für Informatik entscheiden, Anton, dann werde ich es dir gleich tun.“

↑ Wahl des Studienfachs

Die Drillinge Anton, Bernd und Christian stehen davor, sich an der Universität entweder für das Studienfach Mathematik oder für Informatik einzuschreiben, und diskutieren darüber, wer sich für welches Fach entscheiden sollte. Dabei hat jeder der Drei bestimmte Wunschvorstellungen. So verkündet Bernd sofort, dass er Mathematik studieren will, falls einer seiner beiden Brüder sich für Informatik einschreiben sollte. Christian hingegen will Informatik studieren, wenn Bernd dies ebenfalls tut oder wenn Anton sich für Mathematik entscheidet. Um die Angelegenheit noch komplizierter zu machen, mischt sich Anton ein und sagt: „Wenn Bernd Mathe studiert, dann tu ich das auch!“ Daraufhin erwidert Christian: „Solltest du dich aber doch für Informatik entscheiden, Anton, dann werde ich es dir gleich tun.“

A Anton studiert Mathematik, ...

$$\neg A \vee \neg C \rightarrow B$$

$$\neg B \vee A \rightarrow \neg C$$

$$B \rightarrow A$$

$$\neg A \rightarrow \neg C$$

$$A \wedge B \wedge \neg C$$

↑ Prüfziffer-Aufgabe

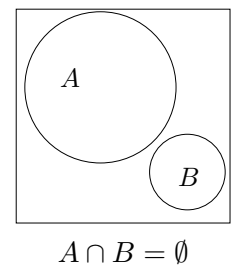
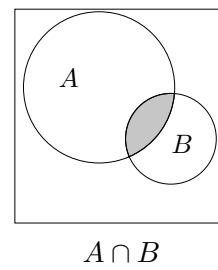
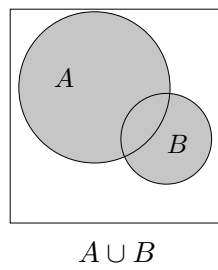
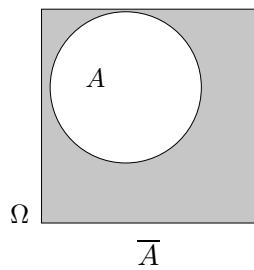
Es ist denkbar und kommt auch häufiger vor, dass bei der Übertragung einer Zahl (wir betrachten hier dreiziffrige Dualzahlen) von einem Speicherplatz zu einem anderen eine Ziffer 1 in eine 0 oder umgekehrt verändert wird. Um diese Fehler zu erkennen, wird zusätzlich zu den Stellen der Zahl eine weitere Stelle als Prüfziffer (auch Prüfbit oder Paritätsbit genannt) eingeführt. Hier wird eine 0 vermerkt, wenn die Anzahl der Ziffern 1 in der Zahl gerade (oder null) ist, sonst 1.

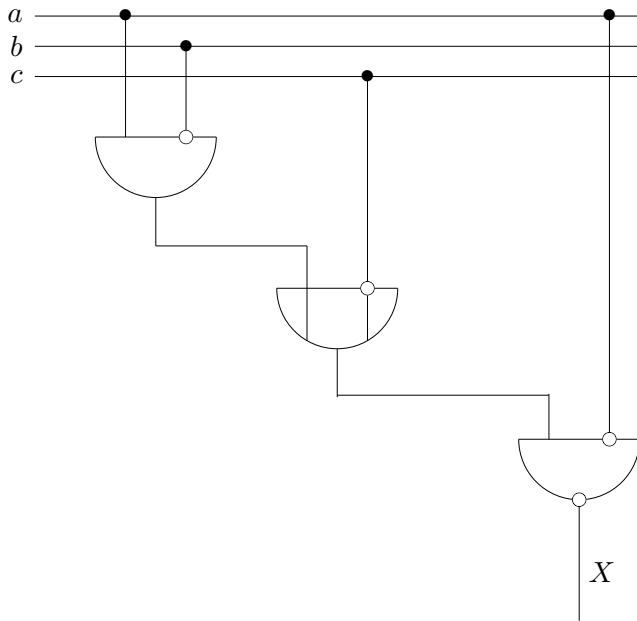
Die Prüfziffer soll automatisch ermittelt werden. Dazu ist eine möglichst einfache Schaltung zu entwerfen.

Die disjunktive Normalform lässt sich nicht vereinfachen.

Gesetze der Mengenalgebra

	1) $A \cap B = B \cap A$ $A \cup B = B \cup A$	Kommutativgesetze
	2) $(A \cap B) \cap C = A \cap (B \cap C)$ $(A \cup B) \cup C = A \cup (B \cup C)$	Assoziativgesetze
\subset enthalten (ist Teilmenge)		
\cup vereinigt	3) $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$	Distributivgesetze
\cap geschnitten	$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$	
\emptyset leere Menge	4) $A \cap (A \cup B) = A$ $A \cup (A \cap B) = A$	Absorptionsgesetze
	5) $\overline{A \cap B} = \overline{A} \cup \overline{B}$ $\overline{A \cup B} = \overline{A} \cap \overline{B}$	De Morgan-Gesetze
	6) $A = \overline{\overline{A}}$ $A \cap \overline{A} = \emptyset$ $A \cup \overline{A} = \Omega$	Komplementgesetze





Ermittle den booleschen Term für X und vereinfache ihn soweit wie möglich.

$$X = a \vee c$$

Wir wollen eine Schaltung mit drei Eingängen konstruieren, an deren Ausgang genau dann der Zustand 1 auftritt, wenn an mindestens zwei Eingängen 1 anliegt. Ein mögliches Anwendungsbeispiel einer solchen Schaltung ist eine Tresortür, die sich nur öffnet, wenn mindestens zwei von drei Schlössern geöffnet werden. Ermittle die disjunktive Normalform und vereinfache sie.

$$X = (\bar{a} \wedge b \wedge c) \vee (a \wedge \bar{b} \wedge c) \vee (a \wedge b \wedge \bar{c}) \vee (a \wedge b \wedge c)$$

$$X = (b \wedge c) \vee (a \wedge c) \vee (a \wedge b)$$

Mathematik

Informatik