

1) a) Coloring: {1: '(B)', 2: '(B)', 3: '(R)', 4: '(B)', 5: '(R)', 6: '(R)', 7: '(R)', 8: '(B)'}

- $1(B) + 2(B) = 3(R)$
- $1(B) + 3(R) = 4(B)$
- $1(B) + 4(B) = 5(R)$
- $1(B) + 5(R) = 6(R)$
- $1(B) + 6(R) = 7(R)$
- $1(B) + 7(R) = 8(B)$
- $2(B) + 3(R) = 5(R)$
- $2(B) + 4(B) = 6(R)$
- $2(B) + 5(R) = 7(R)$
- $2(B) + 6(R) = 8(B)$
- $3(R) + 4(B) = 7(R)$
- $3(R) + 5(R) = 8(B)$

b) inputToDemacs(9) Python code

Fazit: Ab 9 Zahlen ist es unsatisfiable!

Code ist weiter unten im Dokument.

4) Zu zeigen ist, dass die Folge $\varphi \vdash \psi$ valid ist genau unter der Bedingung, dass die Formel $\varphi \rightarrow \psi$ ein Theorem ist. Folgende zwei Bedingungen müssen gezeigt werden.

1. Gilt die Folge $\varphi \vdash \psi$, so ist die Formel $\varphi \rightarrow \psi$ ein Theorem.
2. Wenn die Formel $\varphi \rightarrow \psi$ ein Theorem ist, dann gilt die Folge $\varphi \vdash \psi$.

Für die erste Bedingung müssen wir zeigen, dass jede True Anweisung von φ auch für ψ gilt. Wir nehmen an, dass die Bedingung $\varphi \rightarrow \psi$ kein Theorem ist. Daraus folgt, dass es eine True Anweisung in φ gibt welche nicht ψ zutrifft. Allerdings durch die Bedingung, dass $\varphi \vdash \psi$ gilt, kann so ein Fall nicht eintreten. \Rightarrow wir haben einen Widerspruch. Daher gilt die Aussage, dass $\varphi \rightarrow \psi$ kein Theorem ist nicht!

Die zweite Bedingung besagt, dass wenn die Formel $\varphi \rightarrow \psi$ ein Satz ist, die Folge $\varphi \vdash \psi$ gelten muss. Wenn $\varphi \rightarrow \psi$ ein Theorem ist kann man daraus schlussfolgern, dass jeder True wert, der $\varphi \rightarrow \psi$ erfüllt, auch φ und ψ erfüllt. Da allerdings $\varphi \rightarrow \psi$ äquivalent mit $\varphi \vdash \psi$ ist, wissen wir dass jeder True Wert der für φ gilt auch für ψ gelten muss, was wiederum bedeutet, dass $\varphi \vdash \psi$ valid ist.

5) coloring: 1 = B, 2 = B, 3 = R, 4 = B,

2) a)

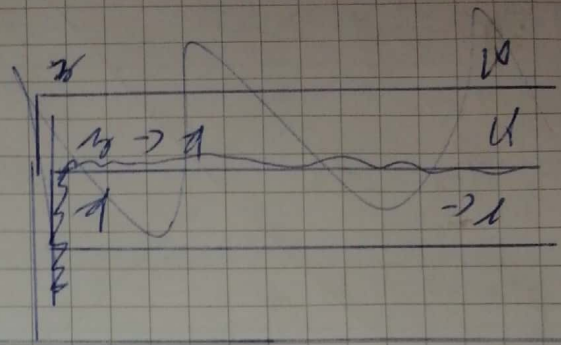
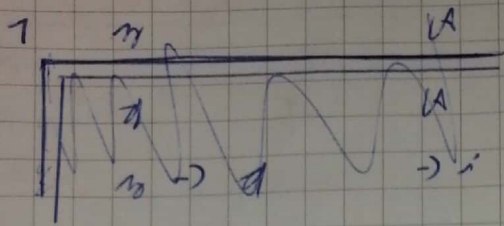
1	$r \rightarrow q$	minimale
2	$r \rightarrow r$	annehmen
3	r	annehmen
4	r	\rightarrow 2, 3
5	q	\rightarrow 1, 4
6	$r \rightarrow q$	\geq 2-5

$$(r \rightarrow r) \rightarrow (r \rightarrow q)$$

b)

1	$r \vee r \wedge r q$	minimale
2	$r \vee r$	\rightarrow 1, 1
3	$r \vee q$	\rightarrow 2, 1
4	$r \vee q$	annehmen
5	q	annehmen
6	\perp	\rightarrow 2, 5
7	r	annehmen
8	\perp	\rightarrow 2, 8
9	\perp	\vee 4-8
10	$r(r \vee q)$	\rightarrow 4-9

3) $\neg p \rightarrow ((\neg p \rightarrow q) \rightarrow q)$



4)

1	$\neg p$	\neg
2	$\neg p \rightarrow q$	\neg
3	q	$\rightarrow e_{1,2}$
4	$(\neg p \rightarrow q) \rightarrow q$	$\rightarrow i_{2,3}$
5	$\neg p \rightarrow ((\neg p \rightarrow q) \rightarrow q)$	$\rightarrow i_{1,4}$

5)

1	$\neg p \wedge (p \vee q)$	\neg
2	$\neg p$	$\wedge e_1$
3	$p \vee q$	$\wedge e_2$
4	assume p	
5	\perp	$\neg e_{2,4}$
6	q	$\perp e_4$
7	assume q	
8	q	$\vee e_{3-7}$

$\neg p \wedge (p \vee q) \rightarrow q$

```

1
2 coloring = {}
3
4 #Anzahl der Zahlen angeben -> CNF wird generiert
5 def inputToDemacs(countTo):
6
7     counter = 0
8     for i in range(1, countTo):
9         for j in range(i+1, countTo):
10             summe = i + j
11             if summe < countTo+1:
12                 #print(f"{i} + {j} = {summe}".format(
13                 i=i, j=j, summe=summe))
14                 print(f"{i} {j} {summe} 0".format(i=i
15                 , j=j, summe=summe))
16                 print(f"-{i} -{j} -{summe} 0".format(
17                 i=i, j=j, summe=summe))
18                 counter += 2
19
20     print(f"p cnf {countTo} {counter}".format(countTo
21     =countTo, counter=counter))
22
23 #generiert den Farbcode für die jeweilige Zahl, je
24 nach signum
25 def inputToDict(src):
26     for num in src:
27         if num > 0:
28             coloring.update({abs(num): '(R)'})
29         else:
30             coloring.update({abs(num): '(B)'})
31     print(coloring)
32
33 #gib Liste Lösungen des SAT Solvers an und erhalte
34 den Color Code inklusive aller Rechnungen
35 def demacsToNatural(src):
36     length = len(src)
37
38     inputToDict(src)
39

```

```
36     for i in range(1, length):
37         for j in range(i+1, length):
38             somme = i + j
39
40             if somme < length+1:
41                 if coloring[i] == coloring[j] and
coloring[j] == coloring[somme]:
42                     print("ERROR!")
43
44                 else:
45                     print("{i}{Vali} + {j}{Valj} = {
summe}{Valsumme}".format(i=i, Vali = coloring[i], j=j
, Valj = coloring[j], somme=somme, Valsumme = coloring
[somme]))
46
47 #EXAMPE WITH NUMBERS 1-8
48 #inputToDemacs(8)
49 #demacsToNatural((-1, -2, 3, -4, 5, 6, 7, -8))
```