Diskrete Strukturen für Informatik

2. Übungszettel

Juliana Hamp Zuniga und Tabea Wittkowski

Schlechter Kuchen kann auch nicht billig, also teuer sein $\Rightarrow \neg a \not\equiv b$.

1.

a) Die Aussage „Guter Kuchen ist nicht billig“ lässt sich logisch als a≡b ausdrücken. Wenn wir beide Aussagen negieren, bleiben sie äquivalent und wir erhalten ¬a≡¬b. Wenn wir nun wieder die sprachlichen Ausdrücke einsetzen, erhalten wir den Satz „Nicht guter Kuchen ist nicht nicht billig“. Wenden wir das Boolesche Gesetz der doppelten Negation an, so lautet der Satz „Nicht guter Kuchen ist billig“, dies wiederum ist tatsächlich inhaltlich äquivalent zu der Aussage „Billiger Kuchen ist nicht gut“.

Es handelt sich leider nicht, um $a \equiv b$. (Implikation)

2/4

b) i) $r \wedge \neg p$ ✓

ii) $(r \wedge p) \rightarrow q$ ✓

iii) $\neg r \rightarrow \neg q$ ✓

2.

a) t1 := $b \wedge (avc)v(a \wedge c)$

a	b	c	avc	bvd	$a \wedge c$	$b \wedge (avc)v(a \wedge c)$
0	0	0	0	0	0	0
0	0	1	1	0	0	0
0	1	0	0	0	0	0
0	1	1	1	1	0	1
1	0	0	1	0	0	0
1	0	1	1	0	1	1
1	1	0	1	1	0	1
1	1	1	1	1	1	1

✓

Term ist erfüllbar. ✓

t2 := $(\neg a \wedge b)v(\neg cvb)$

a	b	c	$\neg a \wedge b$	$\neg cvb$	$(\neg a \wedge b)v(\neg cvb)$ $\equiv \neg c \vee b$
0	0	0	0	~~0~~ 1	~~0~~ 1
0	0	1	0	0	0
0	1	0	1	1	1
0	1	1	1	1	1
1	0	0	0	~~0~~ 1	~~0~~ 1
1	0	1	0	0	0
1	1	0	0	1	1
1	1	1	0	~~0~~ 1	~~0~~ 1

Term ist erfüllbar.

-1

t3 := $((a \wedge b)v(b \wedge c))v(a \wedge c)$

a	b	c	$a \wedge b$	$b \wedge c$	$a \wedge c$	$(a \wedge b)v(b \wedge c)$	$((a \wedge b)v(b \wedge c))v(a \wedge c)$
0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	1	0	0	0	0	0	0
0	1	1	0	1	0	1	1
1	0	0	0	0	0	0	0
1	0	1	0	0	1	0	1
1	1	0	1	0	0	1	1
1	1	1	1	1	1	1	1

Term ist erfüllbar.

t4 := $(a \wedge b)v((cva)v\neg a)$

a	b	c	$a \wedge b$	cva	$\neg a$	$((cva)v\neg a)$	$(a \wedge b)v((cva)v\neg a)$
0	0	0	0	0	1	1	1
0	0	1	0	1	1	1	1
0	1	0	0	0	1	1	1
0	1	1	0	1	1	1	1
1	0	0	0	1	0	1	1
1	0	1	0	1	0	1	1
1	1	0	1	1	0	1	1
1	1	1	1	1	0	1	1

Tautologie

7/8

b) t1≡t3 (Terme 1 und 3 sind logisch äquivalent, da ihre Wahrheitstabellen gleich auswerten.)

Konnte man auch umformen

2/2

3.

a) $(p_1 \wedge p_2) \vee (p_1 \wedge p_3) \vee (p_1 \wedge p_4) \vee (p_2 \wedge p_3) \vee (p_2 \wedge p_4) \vee (p_3 \wedge p_4)$ 3/3

b) $(\neg p_1 \wedge \neg p_2 \wedge p_3 \wedge p_4) \vee (\neg p_1 \wedge p_2 \wedge \neg p_3 \wedge p_4) \vee (\neg p_1 \wedge p_2 \wedge p_3 \wedge \neg p_4) \vee (p_1 \wedge \neg p_2 \wedge \neg p_3 \wedge p_4) \vee (p_1 \wedge \neg p_2 \wedge p_3 \wedge \neg p_4) \vee (p_1 \wedge p_2 \wedge \neg p_3 \wedge \neg p_4)$ 3/3

Super

c) Um allgemein einen logischen Ausdruck für die Aussage „Genau k der Aussagen p1, p2, p3, . . . , pn sind wahr.“ zu bilden würde man wie folgt vorgehen: Die Aussagen p1, p2, p3, . . . , pn werden durch eine Konjunktion („und“/∧) verbunden. Dabei wird die Anzahl n-k der Aussagen durch die Nutzung der Negation („nicht“/¬) verneint, sodass k der Aussagen innerhalb einer Klammer wahr sein müssen, damit der Term insgesamt als wahr ausgewertet werden kann. Dieser Term wird in Klammern gesetzt und mit dem nächsten durch eine Disjunktion („oder“/∨) verbunden. Alle möglichen Kombinationen von n-k negierten und k nicht negierten Aussagen werden nach diesem Muster gebildet, $\binom{n}{k}$ mal.

Genial, wirklich super

Eine andere Möglichkeit wäre das aus a) zu recyceln.

17/20