Benedikt, Marcel, Felix

Aufgabe 2:

Behauptung: Jede Boolesche Funktion f(x,y):{0,1}x{0,1}->{0,1} kann als geschachtelter Ausdruck, bestehend nur aus nand's oder nor's dargestellt werden.

Beweis: Zuerst versuchen wir alle Booleschen Funktionen der Form aus der Behauptung aufzulisten und dann durch gegenseitige Definitionen auf eine kleinere Anzahl an Funktionen zu verringern. Diese werden wir dann durch Definitionen auf nand's und nor's zurueckfuehren.
Dazu betrachten wir die 16 verschiedenen Funktionen:

a	b	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_15	f_14	f_13	f_12	f_11	f_10	f_9	f_8
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Durch Einfuehren der Negation **not**:{0,1}->{0,1}, 0|->1, 1|->0 und der Definition
(**not** f_m)(x):=**not**(f_m(x)) ,fuer alle 0 <= m <= 15, erhalten wir folgende Symmetrie fuer unsere Funktionen:
not f_m = f_m-8, fuer alle 8 <= m <= 15. (I.)

Weiter definieren wir: a **and** b := f_1(a,b). (Konjunktion) (II.)

Nun koennen wir die restlichen Funktionen auf eine Verkettung von **not** und **and** zurueckfuehren.
Aus Gruenden der Vereinfachung definieren wir weiter:
a **or** b := f_7(a,b) = **not** ((**not** a) **and** (**not** b)). (Disjunktion)

Es gilt: f_0(a,b) = a **and** (**not** a) ; (Kontradiktion)
f_2(a,b) = a **and** (**not** b);
f_3(a,b) = f_1(a,b) **or** f_2(a,b);
f_4(a,b) = (**not** a) **and** b;
f_5(a,b) = f_4(a,b) **or** f_1(a,b);
f_6(a,b) = f_4(a,b) **or** f_2(a,b);

Damit konnten wir nun zeigen, dass sich jede binaere boolesche Funktion auf Verschachtelungen der Funtionen **and** und **not** zurueckfuehren laesst.
Es bleibt zu zeigen, dass sich auch diese beiden Funktionen auf Verschachtelungen der Funktion **nand** oder der Funktion **nor** zurueckfuehren lassen.

Zuerst definieren wir **nand**: a **nand** b := f_9(a,b); (III.)

Nun betrachten wir folgende Tabelle und erkennen, dass gilt: **not** a = **nand**(a,a) (IV.)

a	**not**(a)	**nand**(a,a)
1	0	0
0	1	1

Weiterhin folgt aus (I.), (II.) und (III.), dass gilt: **not**(a **nand** b) = a **and** b. (V.)
Damit laesst sich nach (IV.) die Funktion **and** auch auf Verschachtelung der Funktion **nand**

zurueckfuehren.

Nun bleibt nur noch zu zeigen, dass **nor** die Funktion **nand** ersetzen kann.

Wir definieren vorerst: a **nor** b = f_15(a,b);

Analog zu (IV.) gilt: **not** a = a **nor** a; (VI.a)

Aus (I.) folgt: **not** (a **nor** b) = f_7(a,b) = a **or** b; (VI.b)

Es gilt nach (V.): a **nand** b = **not**(a **and** b) = **not** ((**not** a) **or** (**not** b)); (VII.)

Nach (VI.a) und (VI.b) laesst sich der Ausdruck aus (VII.) auf eine Verschachtelung nur aus **nor** Funktionen zurueckfuehren.

q.e.d.