Loogikafunktsioonide süsteemi täpne ja heuristiline minimeerimine – ülesanded ja näidislahendused

1. Loogikafunktsiooni täpne minimeerimine

Esimene näide: On antud funktsioon $\mathbf{f = \bar{a}\,\bar{d} + \bar{a}\,b + a\,\bar{b} + a\,\bar{c}\,d}$ ja see tuleb minimeerida mõne täpse meetodiga. Kasutame selleks McCluskey’ meetodit. Selleks tuleks eelnevalt kirja panna kõik mintermid 1-de piirkonna jaoks. Lahenduskäik on toodud kahe erineva esitusviisi jaoks – intervallid ja numbriline. Mõlemal esitusviisi jaoks kasutame sama muutujate järjestust – **a**, **b**, **c** ja **d**, s.t. **a** vastab vanimale järgule ja **d** vastab noorimale järgule sisendkombinatsioonis. Minimeerimis-algoritm ise koosneb kahest põhietapist - 1) kõikide lihtimplikantide leidmine ja 2) minimaalse lihtimplikant-katte leidmine. Osaliselt määratud funktsioonide korral võetakse lihtimplikantide leidmisel arvesse ka määramata väljundkombinatsioonid (mintermid), sest mõned lihtimplikandid minimaalses kattes võivad katta nii 1-d kui ka määramatusi. Implikant-katte leidmisel vaadeldakse ainult 1-d, sest need lihtimplikandid, mis katavad ainult määramata kombinatsioone, võib kattest välja jätta.

1) Intervall-esitus.
Kõigepealt tuleb leida konjunktsioonidele vastavad intervallid: $\bar{a}\,\bar{d}$ - ‘0--0’, $\bar{a}$ b - ‘01--’, a $\bar{b}$ - ‘10--’ ja a $\bar{c}$ d - ‘1-01’. Intervallid katavad 1-d järgmiselt (mõned 1-d võivad olla kaetud rohkem kui ühe intervalli poolt): ‘0--0’ → ‘0000’, ‘0010’, ‘0100’, ‘0110’; ‘01--’ → ‘0100’, ‘0101’, ‘0110’, ‘0111’; ‘10--’ → ‘1000’, ‘1001’, ‘1010’, ‘1011’; ja ‘1-01’ → ‘1001’, ‘1101’. Saadud 1-de intervallid grupeeritakse 1-de arvu järgi: grupp 0 – ‘0000’; grupp 1 – ‘0010’, ‘0100’, ‘1000’; grupp 2 – ‘0101’, ‘0110’, ‘1001’, ‘1010’; ja grupp 3 – ‘0111’, ‘1011’, ‘1101’.
Järgnevalt kleebitakse kokku need intervallid naabergruppides, mis erinevad ainult ühes järgus. Näiteks ‘0000’ grupist 0 ja ‘0100’ grupist 1 erinevad b-le vastavas järgus. Tulemuseks on uus intervall uues tabelis – ‘0-00’ grupis 0 (‘-’ vaadeldakse 0-na grupeerimisel). Samas erinevad ‘0101’ (gr.2) ja ‘1011’ (gr.3) kahes järgus ja neid kokku kleepida ei saa, kuid ‘0101’ saab kleepida ‘1101’ (gr.3), tulemuseks ‘-101’ grupis 2 uues tabelis. Uue tabeli moodustamisel peab veel arvestama ka seda, et ei tekiks dubleerimisi, sest näiteks nelja sisendkombinatsiooni katvaid intervalle saab kahest erinevast kaheste intervallide paarist moodustada. Kaetud (e. kasutatud) intervallid märgistatakse (‘*’ tabelites). Kui mõni intervallidest jääb katmata, siis on tegu liht-implikandiga ja ta tähistatakse eraldi (suurtähed tabelites). Pärast esimest etappi on moodustunud kõikvõimalikud intervallid (implikandid), mis katavad täpselt kaks sisendkombinatsiooni. Igal juhul tuleks iga etapi lõpus kontrollida, kas mõni kleepimisel kasutamata jäänud intervallidest pole ehk kaetud mõne uue implikandi poolt (kuigi sellist olukorda ei tohiks korrektse kleepimise käigus tekkida!).
Pärast teist etappi, mille käigus kleebitakse implikandid, mis on saadud pärast 1. etappi, saadakse kõikvõimalikud implikandid, mis katavad täpselt neli sisendkombinatsiooni (4-ne kontuur Karnaugh’ kaardis). Antud juhul jäi katmata kaks intervalli – A (‘-101’) ja B (‘1-01’). Saadud tabel ostub antud juhul lõplikuks, sest seal olevaid intervalle enam kokku kleepida ei õnnestu. Seega saime juurde neli lihtimplikanti: C (‘0--0’), D (‘-0-0’), E (‘01--’) ja F (‘10--’). Tabelite kõrval toodud Karnaugh’ kaardil on toodud kõik kuus lihtimplikanti.

mintermid

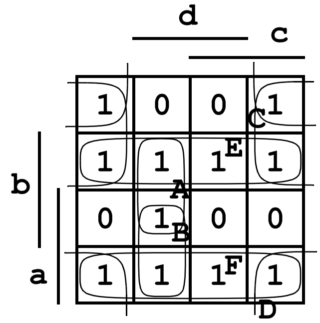
gr.	abcd	
0	0000	*
1	0010	*
	0100	*
	1000	*
2	0101	*
	0110	*
	1001	*
	1010	*
3	0111	*
	1011	*
	1101	*

1. etapp

gr.	abcd	
0	00-0	*
	0-00	*
	-000	*
1	0-10	*
	-010	*
	010-	*
	01-0	*
	100-	*
	10-0	*
2	01-1	*
	-101	A
	011-	*
	10-1	*
	1-01	B
	101-	*

2. etapp

gr.	abcd	
0	0--0	C
	-0-0	D
1	01--	E
	10--	F

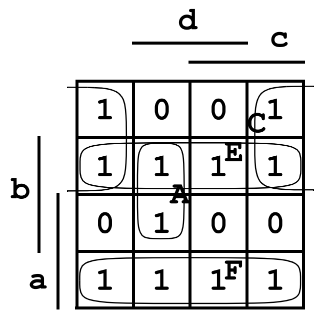


Pärast lihtimplikantide leidmist tuleks genereerida katte leidmiseks lihtimplikantide (veerud) ja nende poolt kaetud 1-de (read) tabel. Saadud tabelis on näha, et implikandid E ja F on ainsad, mis katavad mingit sisendkombinatsiooni (vastavalt '0111' ja '1011'). Seega peavad E ja F igal juhul esinema lõplikus kattes ning nad võib välja jätta edasisest analüüsist. Samuti tuleks eemaldada kõik need sisendkombinatsioonid, mis on vastavate implikantide poolt kaetud. Lihtsustatud tabelis on alles jäänud neli lihtimplikanti, mis peavad katma kolme sisend-kombinatsiooni. On näha, et tabelis on kaks sõltumatut osa – '1101' kaetud kas A või B poolt ning '0000' ja '0010' kaetud C või D poolt. Lõplikuks lahendiks sobib neli ekvivalentset kombinatsiooni – A-C, A-D, B-C või B-D. Valides E-le ja F-le lisaks veel A ja C saame lõplikuks katteks '01--', '10--', '-101' ja '0--0', mis annab $\mathbf{f = \bar{a}\, b + a\, \bar{b} + b\, \bar{c}\, d + \bar{a}\, \bar{d}}$.

lihtimplikantide tabel

	abcd	A	B	C	D	E	F
	0000			+	+		
	0010			+	+		
	~~0100~~			~~+~~		~~+~~	
	~~1000~~				~~+~~		~~+~~
	~~0101~~	~~+~~				~~+~~	
	~~0110~~			~~+~~		~~+~~	
	~~1001~~		~~+~~				~~+~~
	~~1010~~				~~+~~		~~+~~
*	0111					*	
*	1011						*
	1101	+	+				

abcd	A	B	C	D
0000			*	+
0010			*	+
1101	*	+		



2) Numbriline esitus

Analoogselt tuleks kõigepealt leida konjunktsioonidele vastavad intervallid: $\bar{a}\,\bar{d}$ - '0--0', $\bar{a}$ b - '01--', a $\bar{b}$ - '10--' ja a $\bar{c}$ d - '1-01'. Seejärel tuleb leida kümnendekvivalendid (2-nd koodis) kaetud 1-dele: '0--0' → 0, 2, 4, 6; '01--' → 4, 5, 6, 7; '10--' → 8, 9, 10, 11; ja '1-01' → 9, 13. Saadud arvud grupeeritakse samuti 1-de arvu järgi nende 2-nd koodis: grupp 0 - 0; grupp 1 - 2, 4, 8; grupp 2 - 5, 6, 9, 10; ja grupp 3 - 7, 11, 13.

Järgnevalt kleebitakse kokku need väärtused naabergruppides, mis erinevad teinetesiest 2 astme võrra (vastab 2-nd koodi erinevusele ühes järgus). Näiteks 0 grupist 0 ja 4 grupist 1 erinevad 4 võrra ($4=2^2$). Tulemuseks on uus väärtus uues tabelis – '0 (4)' grupis 0 ('(4)' tähistab vahet). Samas erinevad 5 (gr.2) ja 11 (gr.3) 6 võrra, mis ei ole kahe aste, kuid 5 saab kleepida 13 (gr.3), sest vahe on 8, tulemuseks '5 (8)' grupis 2 uues tabelis. Analoogselt tuleks uue tabeli moodustamisel vältida dubleeringute tekkimist. Kaetud koodid märgistatakse (‘*’ tabelites). Kui mõni koodidest jääb katmata, siis on tegu liht-implikandiga ja ta tähistatakse eraldi (suurtähed tabelites). Pärast esimest etappi on moodustunud kõikvõimalikud implikandid, mis katavad täpselt kaks sisendkombinatsiooni.

Teise etapi käigus kleebitakse implikandid, mis on saadud pärast 1. etappi. Lisatingimuseks on see, et sulgudes olevad osad peavad kokku langema. Näiteks ‘0 (2)’ ja ‘4 (2)’ saab kokku kleepida, kuid mitte ‘2 (8)’ ja ‘6 (1)’. Moodustunud implikantidel vastavad 4-stele kontuuridele Karnaugh’ kaardis. Katmata jäi kaks intervalli – A (‘’) ja B (‘’). Uus tabel ostub jälle lõplikuks, sest seal olevaid implikante ei õnnestu rohkem kokku kleepida. Juurde tuli neli lihtimplikanti: C (‘0--0’), D (‘-0-0’), E (‘01--’) ja F (‘10--’).

mintermid

gr.		
0	0	*
1	2	*
	4	*
	8	*
2	5	*
	6	*
	9	*
	10	*
3	7	*
	11	*
	13	*

1. etapp

gr.			
0	0	(2)	*
	0	(4)	*
	0	(8)	*
1	2	(4)	*
	2	(8)	*
	4	(1)	*
	4	(2)	*
	8	(1)	*
	8	(2)	*
2	5	(2)	*
	5	(8)	A
	6	(1)	*
	9	(2)	*
	9	(4)	B
	10	(1)	*

kõik väärtused

gr.			
0	0/2	(2)	*
	0/4	(4)	*
	0/8	(8)	*
1	2/6	(4)	*
	2/10	(8)	*
	4/5	(1)	*
	4/6	(2)	*
	8/9	(1)	*
	8/10	(2)	*
2	5/7	(2)	*
	5/13	(8)	A
	6/7	(1)	*
	9/11	(2)	*
	9/13	(4)	B
	10/11	(1)	*

2. etapp

gr.			
0	0	(2,4)	C
	0	(2,8)	D
1	4	(1,2)	E
	8	(1,2)	F

kõik väärtused

gr.			
0	0/2/4/6	(2,4)	C
	0/2/8/10	(2,8)	D
1	4/5/6/7	(1,2)	E
	8/9/10/11	(1,2)	F

Näites toodud tabelid on üks-üheses vastavuses intervall-esituse abil leitud tabelitega. Samuti on identne minimaalse katte leidmine. Veerud “kõik väärtused” kasutab laiendatud tähistust, kus igal intervallil on loetletud nii kõik väärtused kui ka vahed. Selline tähistus võib hõlbustada katte lahendamist, kuid nõuab rohkem kirjutamist.

Karnaugh’ kaarti võib kasutada abivahendina, kuid seda on ebamugav kasutada suure sisendite arvu korral (üle 6 sisendi tavaliselt) ja kahjuks ei ole tagatud ka minimaalse lahendi leidmine.
Põhimõtteliselt sama moodi toimub ka lahendamine 0-de piirkonna jaoks. Erinevus seisneb selles, et tulemuseks on KNK (või funktsiooni inversiooni DNK).

Teine näide: Funktsioon on antud ühtede ja määramatuste kaudu ning see tuleb minimeerida täpselt – $\mathbf{f(x_3,x_2,x_1,x_0)=\Sigma(0,1,8,9,10)_{'1'}(4,6,12,14)_{'-'}}$. Endiselt on kaks põhietappi – a) kõikide lihtimplikantide leidmine ja b) lihtimplikantide minimaalse katte leidmine. Erinevalt eelmisest näitest, tuleks nüüd arvestada ka seda, et määramatused on kasutusel, st., nad tuleks eraldi tähistada, nt. ‘o’-ga vasakul pool implikante (ja sinisega). Samuti tuleks arvestada, et kui kleepuvad kaks implikanti, mis katavad ainult määramatusi, siis selline tähistus peaks ka edasi kanduma. Kuna antud juhul on ülesanne esitatud sisuliselt tõeväärtus-tabelina, on esimeseks sammuks mintermide jagamine gruppideks. Sellele järgneb kleepimine – kõigepealt mintermid implikantideks, mis ei sõltu ühest muutujast, seejärel implikantideks, mis ei sõltu kahest muutujast jne. Sammud on toodud allpool. Karnaugh’ kaart on jälle kasutusel ainult abistavas/ illustreerivas rollis.

$f(x_3,x_2,x_1,x_0)=\Sigma(0,1,8,9,10)_{1},(4,6,12,14)_{-}$

tõeväärtus-tabel

x_3	x_2	x_1	x_0	f
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	-
0	1	0	1	0
0	1	1	0	-
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	-
1	1	0	1	0
1	1	1	0	-
1	1	1	1	0

mintermid

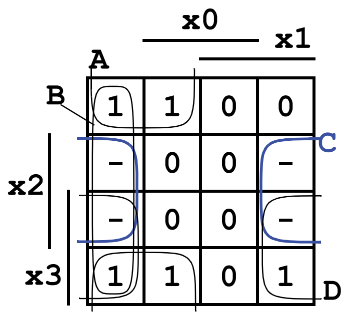
gr.		3210	
0		0000	*
1		0001	*
	o	0100	*
		1000	*
2	o	0110	*
		1001	*
		1010	*
	o	1100	*
3	o	1110	*

1. etapp

gr.		3210	
0		000-	*
		0-00	*
		-000	*
1		-001	*
	o	01-0	*
	o	-100	*
		100-	*
		10-0	*
		1-00	*
2	o	-110	*
		1-10	*
	o	11-0	*

2. etapp

gr.		3210	
0		-00-	A
		--00	B
1	o	-1-0	C
		1--0	D



lihtimplikantide tabel

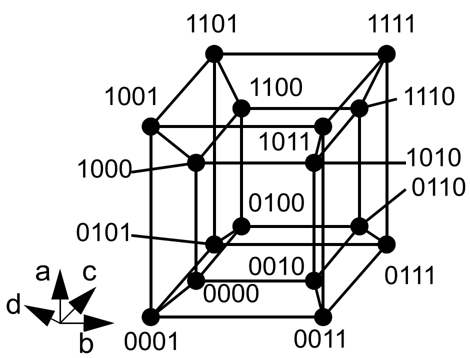
3210	A	~~B~~	D
0000	+	+	
0001	*		
1000	+	+	+
1001	+		
1010			*

Nagu näha, on katab üks moodustunud lihtimplikantidest ainult määramatusi (C). Seega võib selle implikandi katte leidmisel välja jätta, sest määramatusi pole lihtimplikantide tabelisse vaja tuua. Seega peaks tabelis olema ainult 5 mintermi (1-d) ja kolm lihtimplikanti. Kuna B on täielikult kaetud A poolt, siis langeb B välja ja alles jäävad ainult A ja D, mis on ühtlasi ka olulised lihtimplikandid.

Vastus – $\mathbf{f(x_3,x_2,x_1,x_0)= \bar{x}_2\, \bar{x}_1 + x_3\, \bar{x}_0}$.

2. Loogikafunktsiooni heuristiline minimeerimine (Karnaugh kaardi abil)

Esimene ja teine näide: Kuna Karnaugh kaarti on võimalik esitada väga mitmel moel, st. indekseerida võib nii ja teisiti, siis on allpool näidatud kaks varianti – n.ö. tavaline ja appletis kasutatav. Tähelepanu tasub pöörata nii indeksite kui ka muutujate järjekorrale.

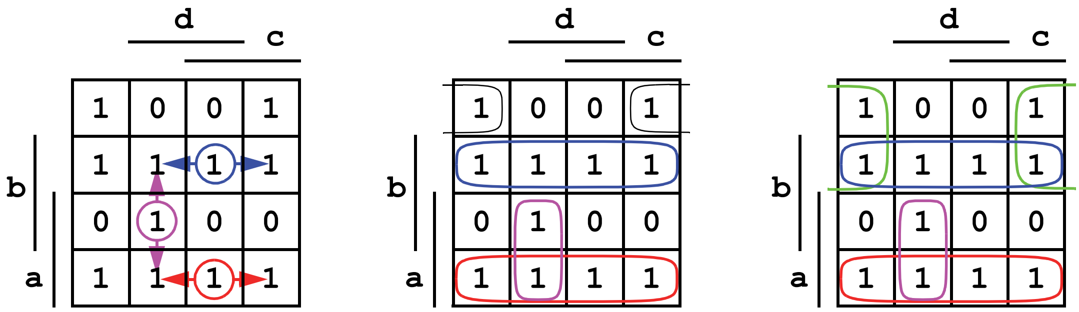


	d	d, b	b	
	0000	0001	0101	0100
c	0010	0011	0111	0110
c, a	1010	1011	1111	1110
a	1000	1001	1101	1100

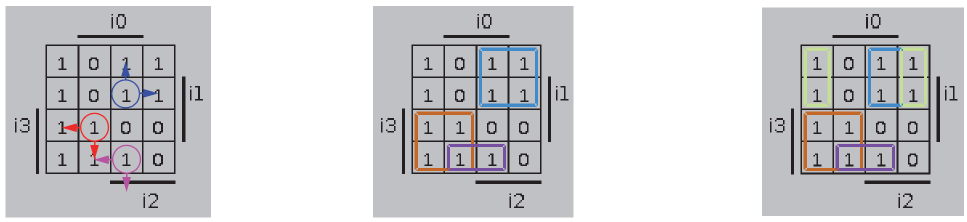
	d	d, c	c	
	0000	0001	0011	0010
b	0100	0101	0111	0110
b, a	1100	1101	1111	1110
a	1000	1001	1011	1010

On antud funktsioon $\mathbf{f(a,b,c,d)=\Sigma(0,2,4,5,6,7,8,9,10,11,13)_{1}}$, ja see tuleb minimeerida heuristiliselt. Alustada on kasulik neist implikantidest, mis on igal juhul vajalikud (vrdl.

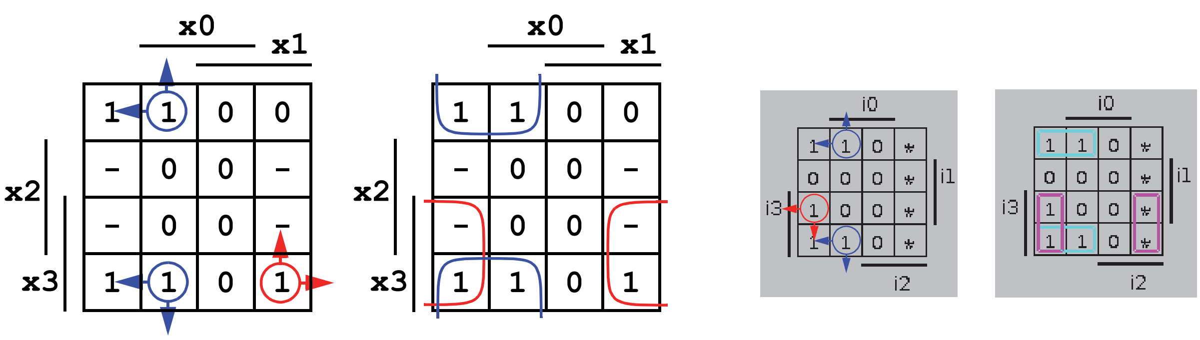
olulised lihtimplikandid). Headeks kandidaatideks on 1-d, mille katmiseks on vähe valikuid – nn. hõredad piirkonnad. Antud näites on sellisteks 1-deks 7 (‘0111’), 11 (‘1011’) ja 13 (‘1101’), mida katvat kontuuri saab laiendada ainult kahes suunas. Esimesed kaks viivad mõlemas suunas laienedes täpselt samale tulemusele – implikandid $\mathbf{\overline{a}b}$ (‘01--’) ja $\mathbf{a\overline{b}}$ (‘10--’). Kolmas võib laieneda nii üles kui ka alla – kas $\mathbf{a\overline{c}d}$ (‘1-01’) või $\mathbf{b\overline{c}d}$ (‘-101’) – valin esimese. Katmata on jäänud ainult 0 (‘0000’) ja 2 (‘0010’), mis koos moodustavad 2-se kontuuri ‘00-0’. See kontuur võib laieneda nii üles kui ka alla – kas $\mathbf{\overline{a}\,\overline{d}}$ (‘0--0’) või $\mathbf{\overline{b}\,\overline{d}}$ (‘-0-0’) – valin teise. Kuna kõik 1-d on kaetud, saab välja kirjutada lõpptulemuse – $\mathbf{f(a,b,c,d) = \overline{a}b + a\overline{b} + a\overline{c}d + \overline{b}\,\overline{d}}$.



Samad sammud appletti kasutades oleksid siis allpool – tulemus täpselt sama, kuigi kontuurid tunduvad erinevat. Muutujad – i3=a, i2=b, i1=c ja i0=d.



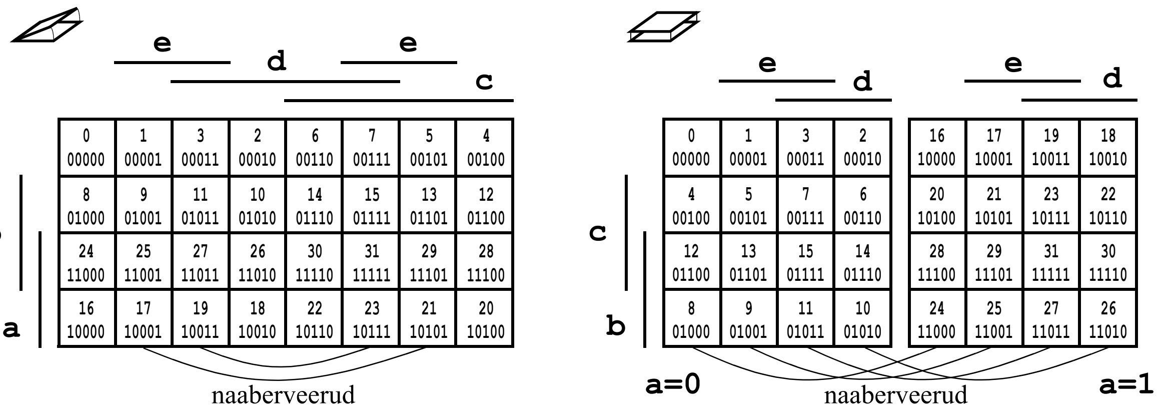
Teine funktsioon on $\mathbf{f(x_3,x_2,x_1,x_0)=\Sigma(0,1,8,9,10)_{‘1’}(4,6,12,14)_{‘\_’}}$. Sellega on veelgi lihtsam, sest need kolm ühte, mis saavad kahes suunas laieneda (‘0001’, ‘1001’ ja ‘1010’), annavad kokku kaks lihtimplikanti – $\mathbf{\overline{x2}\,\overline{x1}}$ (‘-00-’) ja $\mathbf{x3\,\overline{x0}}$ (‘1--0’), millest piisab kõikide 1-de katmiseks. Tulemus seega $\mathbf{f(x_3,x_2,x_1,x_0) = \overline{x2}\,\overline{x1} + x3\,\overline{x0}}$. Kõrval sama tulemus appletiga saadult.

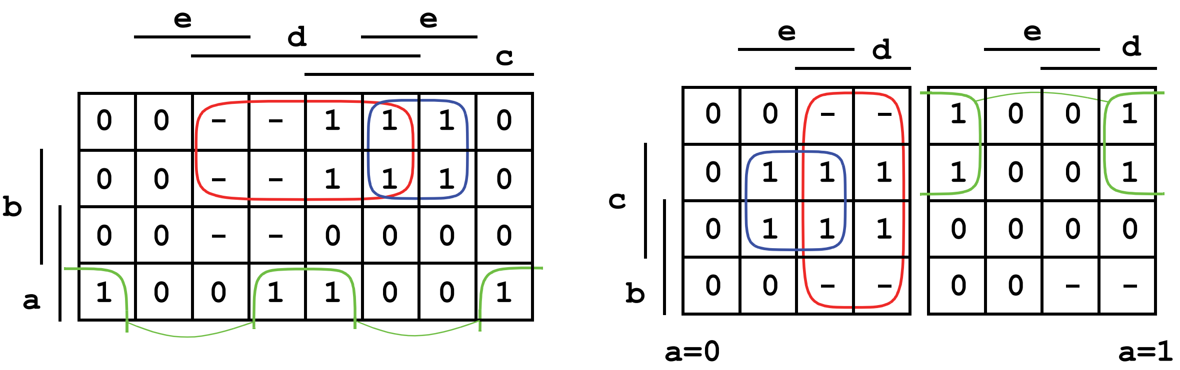


Kolmas näide: Minimeerida Karnaugh’ kaarti kasutades 5-muutja funktsioon, mis on antud 1-de ja määramatus-piirkonna abil:

f(a,b,c,d,e) = Σ(5,6,7,13,14,15,16,18,20,22)‹1› (2,3,10,11,26,27)‹_›.

Viie muutuja korral võib kaardi ehitada kahel moel – ühe kaardina, kus parema ja vasku poole väljad on naabrid telg-sümmeetriliselt (läbi kolmanda mõõtme), ja kahe kaardina, millest üks on mingi muutuja eituse ja teine sama muutuja jaatuse jaoks. Allpool olevad tühjad kaardid illustreerivad võimalusi. Väljadesse on kirjutatud vastavad kümnend- ja kahendväärtused. Tähelepanu tuleks pöörata sellele, millised veerud on omavahel naabrid (lisaks otsmistele ja neile, mis on vahetud naabrid). Kumba neist variantidest kasutada, on maitse ja harjumuse küsimus. Mõlemal juhul on tulemuseks $\mathbf{f = \bar{a}\,d + \bar{a}\,c\,e + a\,\bar{b}\,\bar{e}}$.





1. Loogikafunktsioonide süsteemi täpne minimeerimine

Esimene näide: On antud funktsioonide süsteem (ehk mitme väljundiga funktsioon) ja see tuleb minimeerida mõne täpse meetodiga:

$x(a,b,c) = \bar{a}\,\bar{b}\,\bar{c} + \bar{a}\,b\,\bar{c} + \bar{a}\,b\,c + a\,b\,c$;

$y(a,b,c) = \bar{a}\,\bar{b}\,\bar{c} + \bar{a}\,\bar{b}\,c + \bar{a}\,b\,c + a\,\bar{b}\,c$;

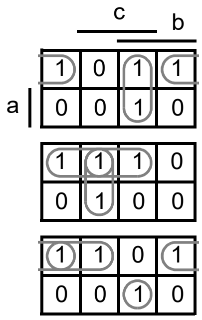
$z(a,b,c) = \bar{a}\,\bar{b}\,\bar{c} + \bar{a}\,\bar{b}\,c + \bar{a}\,b\,\bar{c} + a\,b\,c$.

Kõigepealt teisendatakse funktsioonide süsteem üheks funktsiooniks selliselt, et väljunditele vastab täiendav <u>mitmevalentne</u> sisend (nn. signatuur funktsioon). Saadud funktsioonis on esimesel kolmel sisendil väärtusteks 0 ja 1 ning neljandal sisendil (**e**) on väärtusteks 0, 1 ja 2:

$o(a,b,c,e)=a^0b^0c^0e^0+a^0b^1c^0e^0+a^0b^1c^1e^0+a^1b^1c^1e^0+a^0b^0c^0e^1+a^0b^0c^1e^1+a^0b^1c^1e^1+a^1b^0c^1e^1+$
$+a^0b^0c^0e^2+a^0b^0c^1e^2+a^0b^1c^0e^2+a^1b^1c^1e^2$.

abc	xyz
000	111
001	011
010	101
011	110
100	000
101	010
110	000
111	101

abc	e	o
000	100	1
010	100	1
011	100	1
111	100	1
000	010	1
001	010	1
011	010	1
101	010	1
000	001	1
001	001	1
010	001	1
111	001	1



Kuna täpse minimeerimise korral tuleb alustada mintermidest, siis on ka mitmevalentse sisendi puhul kirja pandud ainult literaalid. Minimeerimine ise on põhimõtteliselt sarnane üksiku Boole'i funktsiooni täpse minimeerimisega - kõigepealt leitakse kõik lihtimplikandid ja seejärel leitakse minimaalne implikantkate. Lihtimplikantide leidmisel kasutatakse samuti kleepimist ja kleepida saab ainult neid implikante, mis on naabrid, st. erinevad ühes sisendi järgus.

Lihtne reegel mitmevalelentse sisendi olemasolu korral oleks järgmine: implikandid erinevad ühes järgus, kui a) nende mitmevalentsed osad on võrdsed ja binaarosad erinevad ühes järgus või b) binaarosad (ja kõik peale ühe MV osa) on võrdsed. Binaarosade kleepimine toimub nii nagu varemgi. Näiteks $a^0b^0c^0e^0$ ('000 *100*') ja $a^0b^1c^0e^0$ ('010 *100*') puhul on MV osad võrdsed ja binaarosa erineb ühes järgus ning kleepimise tulemuseks on $a^0b^{\{0,1\}}c^0e^0$ ('0-0 *100*'). MV osade kleepimisel moodustub nn. hulk-literaal - $a^0b^0c^0e^0$ ('000 *100*') ja $a^0b^0c^0e^1$ ('000 *010*') erinevad ainult MV osas ja kleepimisel saadakse $a^0b^0c^0e^{\{0,1\}}$ ('000 *110*'). Edaspidi on MV-osa puhul kasutusel ainult biti-jada tähistus (näiteks '011'), mis oleks ekvivalentne hulk-literaali tähistusega (vastavalt '{1,2}').

Kahjuks pole implikantide grupeerimine 1-de arvu alusel enam nii lihtne, sest eraldi tuleks arvestada nii binaar- kui ka MV-osa. Praktiliselt võib aluseks võtta ainult binaarosa ja leppida sellega, et MV-osa puhul tuleb teha täiendavat "tühja tööd". Isegi selline grupeerimine lihtsustab kleepuvuse kontrolli - binaarosad saavad kleepuda ainult naabergruppidesse kuuluvate implikantidel ja MV-osad ainult samasse gruppi kuuluvatel implikantidel.

Kleepides esimesena MV osad implikantide vahel, mis kuuluvad gruppi 0, saame kolm uut implikanti - '000 110' ('000 100' ja '000 010'), '000 101' ja '000 011'. Kleepides omavahel gruppidesse 0 ja 1 kuuluvate implikantide binaarosad saame neli uut implikanti - '0-0 100' ('000 100' ja '010 100'), '00- 010', '00- 001' ja '0-0 001'. Saadud seitse uut implikanti moodustavadki uue grupi 0. Sarnaselt on leitud ka ülejäänud uued grupid. Uute implikantide poolt on kaetud kõik mintermid (märgistatud kui '*'). Minimeerides igat funktsiooni üksikuna (vt. Karnaugh' kaardte) oleks z-i puhul lõplikus kattes tegu ühe implikandiga, mis langeks kokku mintermiga ('111'). MV kasutamise korral kataks vastav implikand ka väljundit x ('111 101'). Esimese etapi järel moodustunud 15 implikanti kleebitakse analoogselt - MV osad grupi sees ja binaarosad naaber-guppide vahel. Näiteks '000 110' ja '000 101' kleepumisel moodustunud '000 111' ($a^0b^0c^0e^{\{0,1,2\}}$) sõltub ainult binaarsisenditest ja seda saab interpreteerida selliselt, et kõik väljundid on 1-d, kui sisendis on kombinatsioon '000'. Implikandid A-F jäid katmata ja moodustavad seega osa lihtimplikantidest. Teise etapi järel moodustunud kolm implikanti (G-H) ei kleepu ja moodustavad ülejäänud lihtimplikandid.

Mintermidest ja lihtimplikantidest moodustub lihtimplikantide tabel. Minimaalse implikant-katte leidmine on identne üksiku funktsiooni minimeerimise katte leidmisega. Lihtimpli-kandid

mintermid

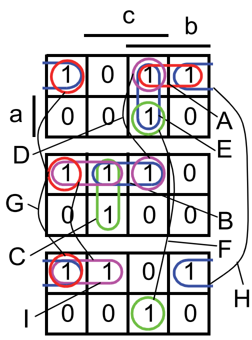
gr.	abc	e	
0	000	100	*
	000	010	*
	000	001	*
1	010	100	*
	001	010	*
	001	001	*
	010	001	*
2	011	100	*
	011	010	*
	101	010	*
3	111	100	*
	111	001	*

1. etapp

gr.	abc	e	
0	000	110	*
	000	101	*
	000	011	*
	0-0	100	*
	00-	010	*
	00-	001	*
	0-0	001	*
1	010	101	*
	001	011	*
	01-	100	A
	0-1	010	B
	-01	010	C
2	011	110	D
	-11	100	E
3	111	101	F

2. etapp

gr.	abc	e	
0	000	111	G
	0-0	101	H
	00-	011	I



lihtimplikantide tabel

	abc	e	A	B	C	D	E	F	G	H	I
	~~000~~	~~100~~							+	+	
	~~000~~	~~010~~							+		+
	~~000~~	~~001~~							+	+	+
	~~010~~	~~100~~	+							+	
	~~001~~	~~010~~		+	+						+
*	001	001									*
*	010	001								*	
	011	100	+			+	+				
	011	010		+		+					
*	101	010			*						
	~~111~~	~~100~~					+	+			
*	111	001						*			

abc	e	A	B	D	E	G
011	100	+		*	+	
011	010		+	*		

abc	e	o
-01	010	1
011	110	1
111	101	1
0-0	101	1
00-	011	1

abc	xyz
-01	010
011	110
111	101
0-0	101
00-	011

C, F, H ja I on olulised, st. ainult nemad katavad mõnda mintermi. Eemaldades need implikandid ja nende poolt kaetud mintermid saame lihtsama tabeli, kust tuleks välja valida ülejäänud vajalikud implikandid. Allesjäänud lihtimplikantidest on D ainus, mis katab kõik ülejäänud mintermid. Lõplik implikantkate koosneb seega implikantidest C, D, F, H ja I, mis annab minimeeritud MV-funktsiooniks

$o(a,b,c,e)=a^{\{0,1\}}b^0c^1e^1+a^0b^1c^1e^{\{0,1\}}+a^1b^1c^1e^{\{0,2\}}+a^0b^{\{0,1\}}c^0e^{\{0,2\}}+a^0b^0c^{\{0,1\}}e^{\{1,2\}}$ ehk

$o(a,b,c,e)=b^0c^1e^1+a^0b^1c^1e^{\{0,1\}}+a^1b^1c^1e^{\{0,2\}}+a^0c^0e^{\{0,2\}}+a^0b^0e^{\{1,2\}}$.

Teisendades minimeeritud MV-funktsiooni tagasi funktsioonide süsteemiks saame

$\mathbf{x(a,b,c) = \bar{a}\,b\,c + a\,b\,c + \bar{a}\,\bar{c}}$; $\mathbf{y(a,b,c) = \bar{b}\,c + \bar{a}\,b\,c + \bar{a}\,\bar{b}}$; $\mathbf{z(a,b,c) = a\,b\,c + \bar{a}\,\bar{c} + \bar{a}\,\bar{b}}$.

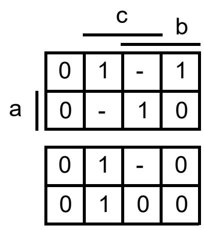
Sealjuures on osa implikante kasutusel kahe väljundi jaoks.

Teine näide: On antud kolme sisendi ja kahe väljundiga loogikafunktsioonide süsteem, mis tuleb minimeerida täpselt. Erinevalt eelmisest näitest on lisandunud ka määramatused – $\mathbf{x(a,b,c)=\Sigma(1,2,7)_{‘1’}(3,5)_{‘\_’}}$ ja $\mathbf{y(a,b,c)=\Sigma(1,5)_{‘1’}(3)_{‘\_’}}$.

Tõeväärtus- ja signatuur-tabelid on toodud paremal koos Karnaugh kaartidega. Nagu üksiku funktsiooni minimeerimisel, peab ka funktsioonide süsteemi minimeerimisel määramatusi käsitlema kui võimalikke väärtusi (1-d antud juhul). Alles implikant-katte leidmisel eemaldatakse määramatused edasisest analüüsist. On loomulik, et ka nüüd tähistatkse määramatused eraldi (‘o’-ga), et implikantide valimisel võiks ainult määramatusi katvad implikandid kohe välja jätta.

abc	xy
000	00
001	11
010	10
011	--
100	00
101	-1
110	00
111	10

abc	e	o
001	10	1
010	10	1
o 011	10	1
o 101	10	1
111	10	1
001	01	1
o 011	01	1
101	01	1

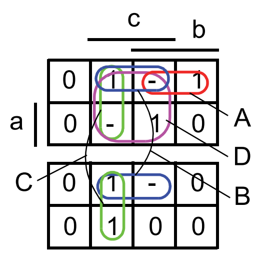


Minimeerimise alustamiseks tuleks sorteerida mintermid, mis ongi just signatuur-tabelis kirjas. Järgneva kleepimise käigus moodustub kokku neli liht-implikanti.

```
mintermid

gr. abc  e
1     001 10 *
      001 01 *
      010 10 *
2 o   011 10 *
  o   011 01 *
  o   101 10 *
      101 01 *
3     111 10 *
```

```
1. etapp

gr. abc  e
1     001 11 *
      0-1 10 *
      -01 10 *
      0-1 01 *
      -01 01 *
      01- 10 A
2 o   011 11 *
      101 11 *
      -11 10 *
      1-1 10 *
```

```
2. etapp

gr. abc  e
1     0-1 11 B
      -01 11 C
      --1 10 D
```



Neist neljast liht-implikandist on olulised kolm – A, C ja D. Sealjuures on B täielikult kaetud C poolt – B katab 1-dest “001 10” ja “001 01”, kuid C katab veel ka “101 01”. Lõpp-tulemust välja kirjutades selgub veel üks huvitav detail. Nimelt ei arvesta minimaalset implikantide arvu leidvad meetodid sellega, kui hõre/tihe saab olema väljundite pool, st. mitu implikanti iga väljund tegelikult vajaks. Antud juhul on C liigne x-i jaoks, sest D katab teda täielikult.

lihtimplikantide tabel

	abc	e	A	B	C	D
	~~001~~	~~10~~		+	+	+
	~~001~~	~~01~~		+	+	
*	010	10	*			
*	101	01			*	
*	111	10				*

abc	e	o
01-	10	1
-01	11	1
--1	10	1

abc	xy
01-	10
-01	11
--1	10

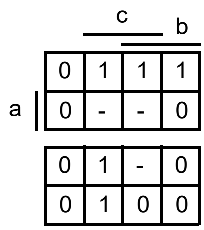
abc	xy
01-	10
-01	01
--1	10

Ehk neeldumis-seadusi arvestades:
Esialgne tulemus: **$x=\bar{a}b+\bar{b}c+c$; $y=\bar{b}c$;**
Lõplik tulemus: **$x=\bar{a}b+c$; $y=\bar{b}c$;**

Kolmas näide: Tegemist on eelmise näite minimaalse modifikatsiooniga – esimeses funktsioonis on üks ‘1’ ja ‘-’ oma kohad ära vahetanud. Seega, on antud järgmine kolme sisendi ja kahe väljundiga loogikafunktsioonide süsteem, mis tuleb minimeerida täpselt – $\mathbf{x(a,b,c)=\Sigma(1,2,3)_{1}(5,7)_{-}}$ ja $\mathbf{y(a,b,c)=\Sigma(1,5)_{1}(3)_{-}}$.
Tõeväärtus- ja signatuur-tabelid on toodud paremal koos Karnaugh kaartidega. Kuna mintermid on täpselt samad, siis tekivad ka täpselt samad neli liht-implikanti. Erinevus on siiski selles, millised implikandid katavad ainult määramatusi (‘o’).

abc	xy
000	00
001	11
010	10
011	1-
100	00
101	-1
110	00
111	-0

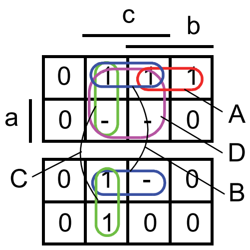
abc	e	o
001	10	1
010	10	1
011	10	1
o 101	10	1
o 111	10	1
001	01	1
o 011	01	1
101	01	1




```
mintermid

gr. abc  e
1   001 10 *
    001 01 *
    010 10 *
2   011 10 *
  o 011 01 *
  o 101 10 *
    101 01 *
3 o 111 10 *
```

```
1. etapp

gr. abc  e
1   001 11 *
    0-1 10 *
    -01 10 *
    0-1 01 *
    -01 01 *
    01- 10 A
2   011 11 *
    101 11 *
    -11 10 *
  o 1-1 10 *
```

```
2. etapp

gr. abc  e
1   0-1 11 B
    -01 11 C
    --1 10 D
```



Neist neljast liht-implikandist on olulised kaks – A ja C. Sealjuures on D täielikult kaetud B poolt – D katab 1-dest “001 10” ja “011 10”, kuid B katab veel ka “001 01”. Samas pole ka B-d vaja, sest A ja C katavad kõik mintermid.

lihtimplikantide tabel

	abc e	A	B	C	D
	~~001 10~~		+	+	+
	~~001 01~~		+	+	
*	010 10	*			
	~~011 10~~	+	+		+
*	101 01			*	

abc	e	o
01-	10	1
-01	11	1

abc	xy
01-	10
-01	11

Lõpp-tulemus seega: $\mathbf{x=\bar{a}b+\bar{b}c;}$ $\mathbf{y=\bar{b}c;}$

Neljas näide: Tegemist on kahe eelmise näite kombineerimisega üheks suuremaks näiteks – esimene väljund on esimesest funktsioonist (vt. teine näide), teine on mõlemal ühine ja kolmas väljund on teise funktsiooni (vt. kolmas näide) esimene väljund. Seega, on antud järgmine kolme sisendi ja kolme väljundiga loogikafunktsioonide süsteem, mis tuleb minimeerida täpselt – $\mathbf{x(a,b,c)=\Sigma(1,2,7)_{1}(3,5)_{-}}$; $\mathbf{y(a,b,c)=\Sigma(1,5)_{1}(3)_{-}}$ ja $\mathbf{z(a,b,c)=\Sigma(1,2,3)_{1}(5,7)_{-}}$. Tõeväärtus-tabel on toodud allpool koos Karnaugh kaartidega. Minterme on selles süsteemis rohkem ja ka kleepumisel tekib oluliselt rohkem implikantide kombinatsioone. Signatuur-funktsioon ja mitme-valentne sisend on jäetud meelega eraldi välja toomata. Selle asemel on näidatud otse väljundeid ning implikantide kleepumisel tuleb arvestada kas sisendite ühejärgulist erinevust väljundite kokkulangemisel (ja selle erineva sisendi asendumist eba-olulise väärtusega) või sisendite täielikul kokku langemisel erinevate väljundite “liitumist” samasse implikanti.

abc	xyz
000	000
001	111
010	101
011	--1
100	000
101	-1-
110	000
111	10-

mintermid

gr.		abc	xyz	
1		001	100	*
		001	010	*
		001	001	*
		010	100	*
		010	001	*
2	o	011	100	*
	o	011	010	*
		011	001	*
	o	101	100	*
		101	010	*
	o	101	001	*
3		111	100	*
	o	111	001	*

1. etapp

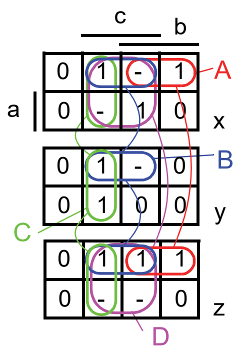
gr.		abc	xyz	
1		001	110	*
		001	101	*
		001	011	*
		010	101	*
		0-1	100	*
		-01	100	*
		0-1	010	*
		-01	010	*
		0-1	001	*
		-01	001	*
		01-	100	*
		01-	001	*
2	o	011	110	*
		011	101	*
		011	011	*
		101	110	*
	o	101	101	*
		101	011	*
		-11	100	*
		-11	001	*
		1-1	100	*
	o	1-1	001	*
3		111	101	*

2. etapp

gr.	abc	xyz	
1	001	110	*
	0-1	110	*
	0-1	101	*
	0-1	011	*
	-01	110	*
	-01	101	*
	-01	011	*
	01-	101	A
2	011	111	*
	101	111	*
	-11	101	*
	1-1	101	*

3. etapp

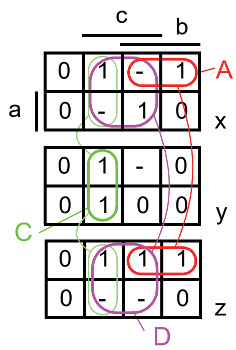
gr.	abc	xyz	
1	0-1	111	B
	-01	111	C
	--1	101	D



Katte leidmine, st. väikseima hulga implikantide leidmine toimub sarnaselt teistele funktsioonidele – implikantide tabelisse pannakse kirja ainult need mintermid, mis katavad ühtesi, tehakse kindlaks olulised lihtimplikandid ja allesjäänutest valitakse väikseim arv implikante selliselt, et kõik mintermid oleksid kaetud. Viimase sammuna toimub väljundite hõredamaks tegemine – implikanti C (‘-01’) pole väljundite *x* ja *z* jaoks vaja, sest vastavad mintermid (‘001’ ja ‘101’) on kaetud juba implikandi D (‘--1’) poolt.

abc	xyz	A	B	C	D
001	100		+	+	+
001	010		+	+	
001	001		+	+	+
010	100	*			
010	001	*			
011	001	+	+		+
101	010			*	
111	100				*

abc	xyz
01-	101
-01	111
--1	101



lõpptulemus

abc	xyz
01-	101
-01	010
--1	101

Lõpp-tulemus seega: $\mathbf{x=\bar{a}b+c;}$ $\mathbf{y=\bar{b}c;}$ $\mathbf{z=\bar{a}b+c\ (=x);}$

2. Funktsioonide süsteemi heuristiline minimeerimine

Heuristilise minimeerimise põhi-idee seisneb selles, et kõikide lihtimplikantide leidmise asemel üritatakse olemasolevaid implikante ümber grupeerida selliselt, et uued implikandid kataksid olemasolevaid. Üldjuhul väheneb sealjuures ka implikantide koguarv. Grupeerimisel arvestatakse kleepimis-reegleid täpselt sama moodi kui täpse minimeerimise korral. Vajaduse korral jagatakse üks implikant kaheks (või enamaks) väiksemaks implikandiks, st. toimub kleepimise vastandtegevus.

Heuristilisel minimeerimisel Karnaugh kaardi abil on põhimõtted samad – kontuure e. implikante tuleb laiendada selliselt, et nad kataksid võimalikult palju teisi kontuure. Alustada tuleks sealjuures sellistest ‘1’-dest (‘0’-dest), mis on ainult ühes funktsioonidest. Mõte on selles, et sellise ‘1’ (‘0’) katmiseks on igal juhul vaja implikanti ja see implikant esineks ainult selles konkreetses funktsioonis. Näideteks sobivad samad funktsioonide süsteemid, mida kasutati täpsel minimeerimisel.

Esimene näide: On antud funktsioonide süsteem, mis tuleb minimeerida heuristiliselt:

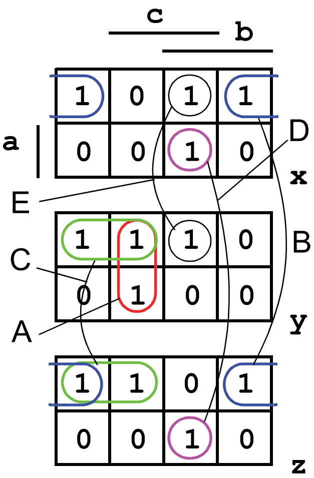
$x(a,b,c) = \bar{a}\,\bar{b}\,\bar{c} + \bar{a}\,b\,\bar{c} + \bar{a}\,b\,c + a\,b\,c\,;$

$y(a,b,c) = \bar{a}\,\bar{b}\,\bar{c} + \bar{a}\,\bar{b}\,c + \bar{a}\,b\,c + a\,\bar{b}\,c\,;$

$z(a,b,c) = \bar{a}\,\bar{b}\,\bar{c} + \bar{a}\,\bar{b}\,c + \bar{a}\,b\,\bar{c} + a\,b\,c\,.$

Funktsioonidel **x** ja **z** ei leidu mitte ühtegi sellist ‘1’, mida mõnes teises funktsioonis ei leiduks. Kuid **y** puhul on olukord teine – “$a\bar{b}c$” esineb ainult sela (vt. ka tõeväärtus-tabelit, kus vastavas reas on ainult üks väljund kaetud. Suurim kontuur, mis seda kataks, oleks A (“$\bar{b}c$”). Kuna rohkem selliseid ‘1’ pole, tuleks hakata otsima kontuuride paare (ja edaspidi kolmikuid jne.). Ehk siis, vaadata, kas ühe funktsiooni mõne kontuuri korral sobiks see ka mõnele teisele funktsioonile. Funktsiooni **x** puhul kolmest kontuurist (“$\bar{a}\bar{c}$”, “$\bar{a}b$” ja “bc”) sobib selleks just “$\bar{a}\bar{c}$” (B), mis on kasutatav ka **z** jaoks. Sama moodi sobib **y** jaoks kahest võimalikust lisakontuurist (“$\bar{a}\bar{b}$” ja “$\bar{a}c$”, sest “$\bar{b}c$” on juba kasutusel) just “$\bar{a}\bar{b}$” (C), sest sobib samuti ka **z** jaoks. Ja **z** ainsana katmata jäänud ‘1’ jaoks peab kasutama kontuuri D (“abc”), mis sobib aga ka **x** jaoks. Kahe viimase ‘1’ katmiseks sobib üks ühine kontuur x ja y jaoks – “$\bar{a}bc$” (E).

abc	xyz
000	111
001	011
010	101
011	110
100	000
101	010
110	000
111	101



abc	xyz
-01	010
0-0	101
00-	011
111	101
011	110

Lõpp-tulemus osutub identseks täpsel minimeerimisel saaduga:

$\mathbf{x = \bar{a}\,b\,c + a\,b\,c + \bar{a}\,\bar{c}\,; \quad y = \bar{b}\,c + \bar{a}\,b\,c + \bar{a}\,\bar{b}\,; \quad z = a\,b\,c + \bar{a}\,\bar{c} + \bar{a}\,\bar{b}\,.}$

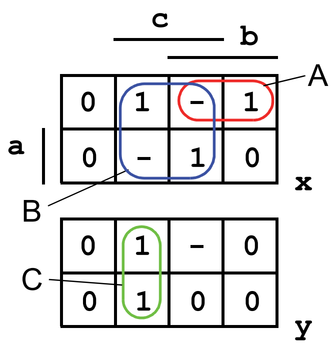
Teine näide: On antud kaks funktsiooni, mis tuleb koos minimeerida:

x(a,b,c)=Σ(1,2,7)‘1’(3,5)‘_’ ja **y(a,b,c)=Σ(1,5)‘1’(3)‘_’** .

Antud juhul on funktsioonil **x** kaks ‘1’, mis puuduvad teises funktsioonis – “$\bar{a}b\bar{c}$” ja “abc”. Neid katvad kontuurid on A (“$\bar{a}b$”) ja B (“c”). Kuna **x** on nüüd täielikult kaetud, võib y sõltumatult minimeerida. Selleks on vaja ainult ühte kontuuri – C (“$\bar{b}c$”).
Ka antud juhul on tulemus ekvivalentne täpsel minimeerimisel saaduga: **x=$\bar{a}$b+c; y=$\bar{b}$c** .

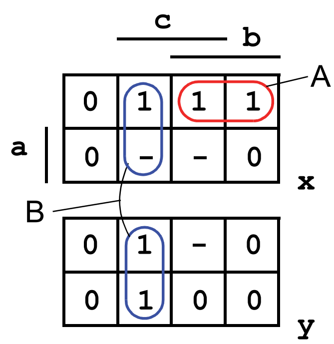
Kolmas näide: On antud kaks funktsiooni, mis tuleb koos minimeerida:
$x(a,b,c)=\Sigma(1,2,3)_{1},(5,7)_{-}$ ja **$y(a,b,c)=\Sigma(1,5)_{1},(3)_{-}$** .
Antud juhul on funktsioonil **x** üks ‘1’, mis puudub teises funktsioonis – “$\bar{a}bc$”. Seda kattev kontuur on A (“$\bar{a}b$”). Kuigi funktsioonil y on ‘1’ (“$a\bar{b}c$”), mis x-l on ‘-’, ei saa seda arvestada, sest määramatused võivad realiseeruda nii ‘1’ kui ‘0’-na. Ülejäänud ühtede katmiseks sobib kontuuri-paar B (“$\bar{b}c$”).
Ka antud juhul on tulemus ekvivalentne täpsel minimeerimisel saaduga: **x=$\bar{a}$b+$\bar{b}$c; y=$\bar{b}$c** .

Teine näide



abc	xy
01-	10
--1	10
-01	01

Kolmas näide



abc	xy
01-	10
-01	11

Lõpuks neljandas loengus toodud nelja sisendi ja väljundiga funktsioonide süsteemi kolm erinevat võimalikku lahendust – esimene on leitud espresso abil, teine appletti kasutades ja kolmas on loengus esitatud lahendus. Kontrollimiseks on võimalik kasutada espressot – parameeter “-Dverify” lubab võrrelda kahte faili/tabelit.

espresso

abcd	klmn
110-	1000
-1-0	0100
1-11	0001
-100	0001
-10-	0010
-001	1101
0-10	1010
101-	0001
00--	0100
-011	1001
10-0	0010

c=11, i=30, o=15

applett

abcd	klmn
110-	1000
-1-0	0100
1-11	0001
-100	0001
-10-	0010
-001	0100
0-10	1110
101-	0011
1-00	0010
00-1	0101
-0-1	1000

c=11, i=30, o=15

loeng

abcd	klmn
110-	1000
-1-0	0100
1-11	0001
-100	0001
-10-	0010
-001	0100
0-10	1110
101-	0001
10-0	0010
00-1	0101
-0-1	1000

c=11, i=30, o=14