TTÜ1918

Kahetasemeline minimeerimine ja mitmetasemeline realisatsioon

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

mintermid

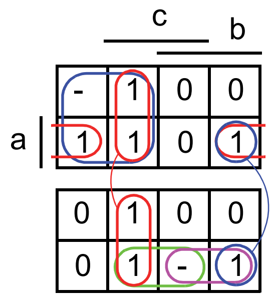
gr.		abc	e
0	o	000	10
1		001	10
		001	01
		100	10
2		101	10
		101	01
		110	10
		110	01
3	o	111	01

1. etapp

gr.	abc	e	
0	00-	10*	
	-00	10*	
1	001	11*	
	-01	10*	
	-01	01*	
	10-	10*	
	1-0	10	A
2	101	11*	
	110	11	B
	1-1	01	C
	11-	01	D

2. etapp

gr.	abc	e	
0	-0-	10	E
1	-01	11	F



lihtimplikantide tabel

	abc	e	A	B	C	D	E	F
	~~001~~	~~10~~					~~+~~	~~+~~
*	001	01						*
	100	10	+				+	
	~~101~~	~~10~~					~~+~~	~~+~~
	~~101~~	~~01~~			~~+~~			~~+~~
	110	10	+	+				
	110	01		+		+		

abc	e	A	B	C	D	E
100	10	+				+
110	10	+	+			
110	01		+		+	

3 varianti

1: F, A, B
2: F, A, D
3: F, B, E

Kahetasemeline minimeerimine ja mitmetasemeline realisatsioon

3 varianti
1: F, A, B
2: F, A, D
3: F, B, E

Lisaks üksikult minimeeritud
4: A, E, D, F

abc	xy
-01	11
1-0	10
110	01

abc	xy
-01	11
1-0	10
11-	01

abc	xy
-01	01
110	11
-0-	10

abc	xy
1-0	10
-0-	10
11-	01
-01	01

c b, a

	-	1	0	0
a	1	1	0	1

0	1	0	0
0	1	-	1

c b, a

	-	1	0	0
a	1	1	0	1

0	1	0	0
0	1	-	1

c b, a

	-	1	0	0
a	1	1	0	1

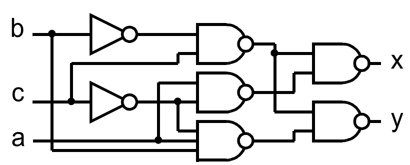
0	1	0	0
0	1	-	1

c b, a

	-	1	0	0
a	1	1	0	1

0	1	0	0
0	1	-	1

bi=b'; ci=c'; t1=bi·c;
t2=a·ci; t3=a·b·ci;
x=t1+t2; y=t1+t3;



NOT - 2
2-NAND - 4
3-NAND - 1
26 transistori
[13 literaali]

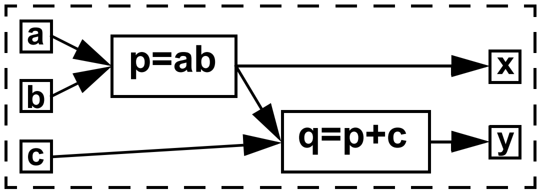
NOT - 2
2-NAND - 5
3-NAND - 0
24 transistori
[12 literaali]

NOT - 2
2-NAND - 3
3-NAND - 1
22 transistori
[11 literaali]

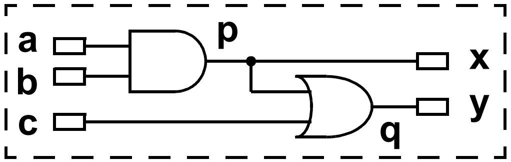
NOT - 2
2-NAND - 5
3-NAND - 0
24 transistori
[12 literaali]

Mitmetasemeline loogikafunktsioonide minimeerimine

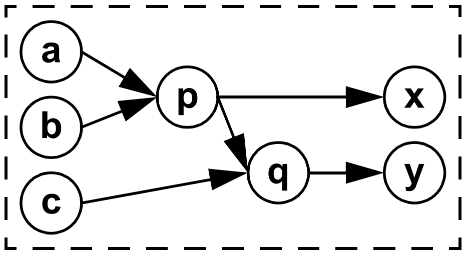
- Loogikaelementide teegid
- Loogikalülid või makrod (PLA)?
  - paindlikkus / jõudlus / ühendused



- Esitusviis – loogikavõrkgraaf (logic network)
  - omavahel ühendatud loogikafunktsioonid
  - kombineeritud struktuurne/käitumuslik mudel
- Seotud võrkgraaf (bound/mapped network)
  - omavahel ühendatud loogikalülid
  - struktuurne mudel



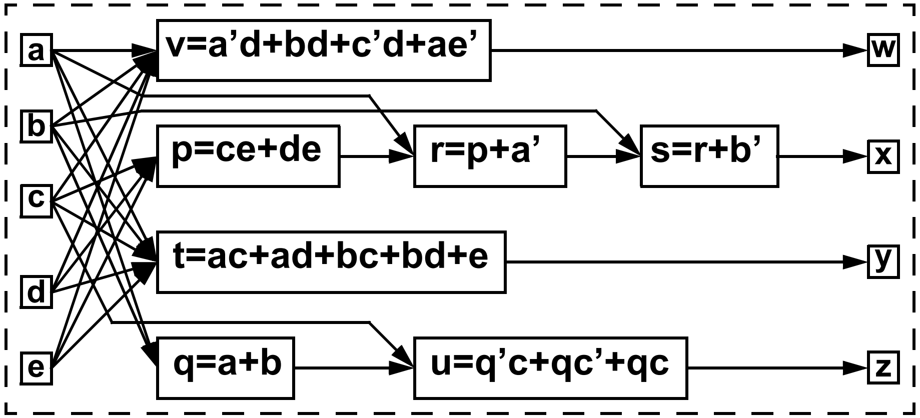
- Eesmärk
  - pindala, viide
  - võimsus, testitavus

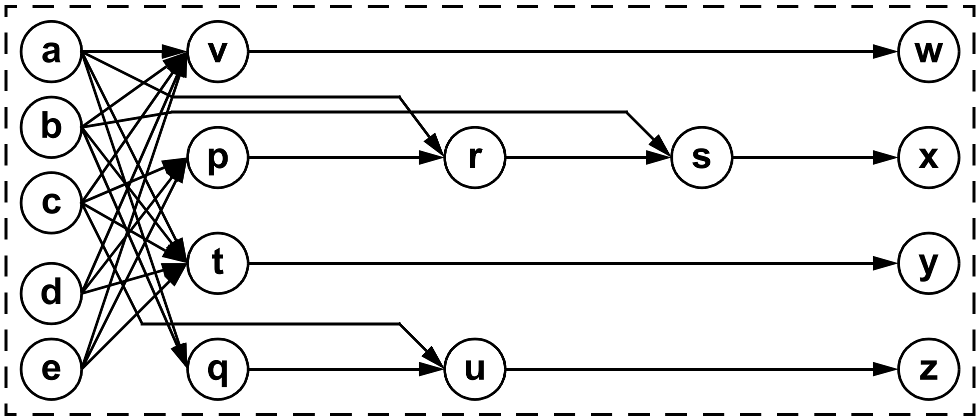


Näitevõrkgraaf

```
p=ce+de
q=a+b
r=p+a'
s=r+b'
t=ac+ad+bc+bd+e
u=q'c+qc'+qc
v=a'd+bd+c'd+ae'
w=v
x=s
y=t
z=u

    w= a'd+bd+c'd+ae'
f:  x= a'+b'+ce+de
    y= ac+ad+bc+bd+e
    z= a+b+c
```





TTÜ 1918

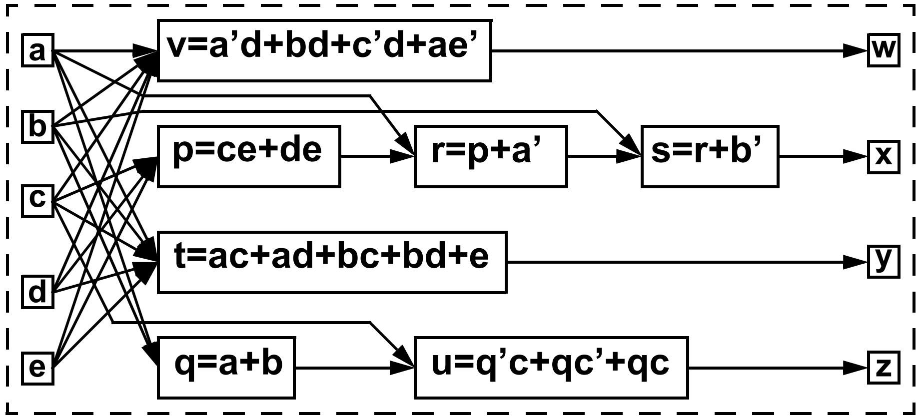
Optimeerimine

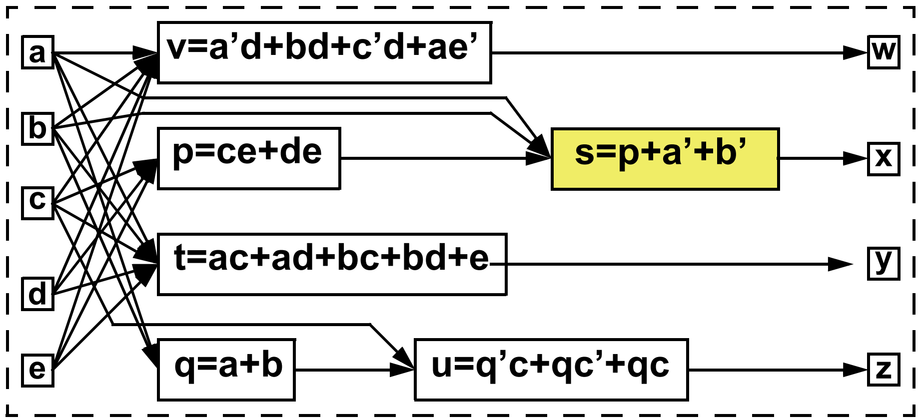
- **Pindala (võimsustarbe) estimatsiooni minimeerimine**
  - **arvestada tuleb viite piiranguid**
- **Suurima viite minimeerimine**
  - **arvestada tuleb pindala (võimsustarbe) piiranguid**
- **Testitavuse maksimeerimine**
- **Võimsustarbe minimeerimine**
- **Estimatsioon (ennustus, estimation)**
  - **Pindala – literaalide arv; funktsioonide/loogikalülide arv**
  - **Viide – tee sügavus; loogikalülide mudelid; olulised teed**
- **Mitmetasemeline minimeerimine on *raske* !**
  - **Täpsed meetodid – eksponentsiaalne keerukus, ebapraktiline**
  - **Ligikaudsed meetodid – heuristilised algoritmid; reeglitel baseeruvad meetodid**
- **Optimeerimis-strateegiad**
  - **Samm-sammuline parendamine – teisendused (transformations) võrkgraafil**
  - **Funktsionaalsus ei tohi muutuda**
  - **Erinevad meetodid – teisenduste variandid & teisenduste rakendamise järjekorrad**

Eemaldamine – Elimination

- *Eemaldatakse* üks funktsioon
- Asendatakse vastavad muutujad

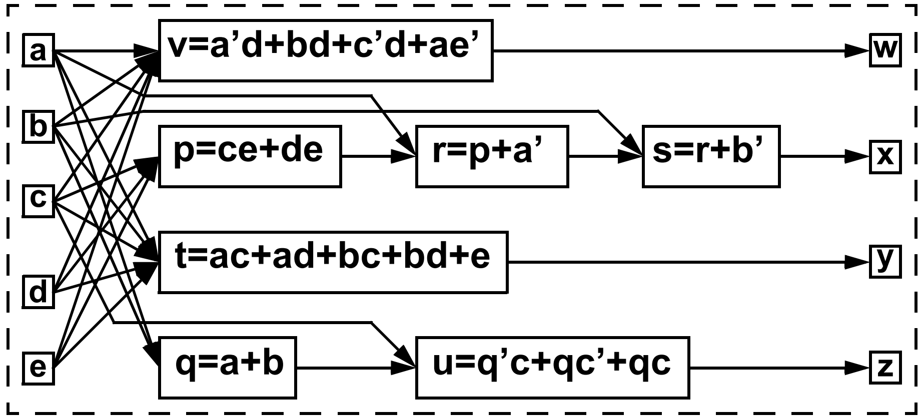
- Näide
  - eemaldatakse *r*
    `s=r+b'; r=p+a';`
  - asendus *s*-s
    `s=p+a'+b';`

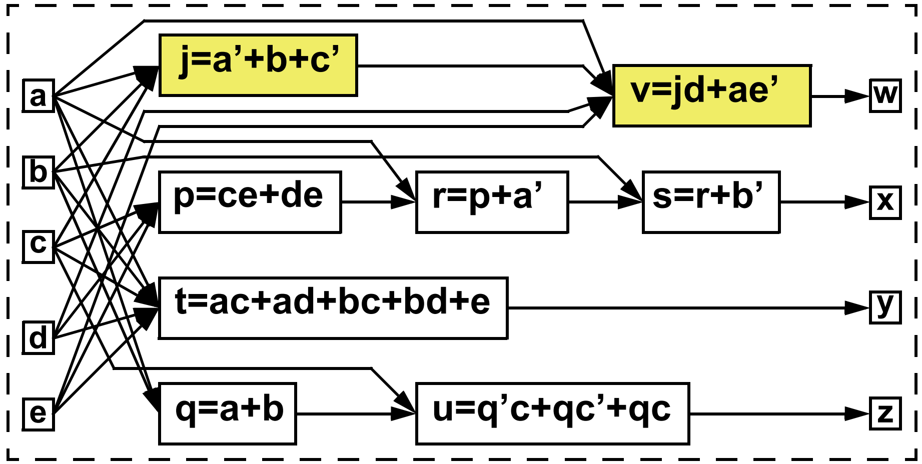




Dekompositsioon – Decomposition

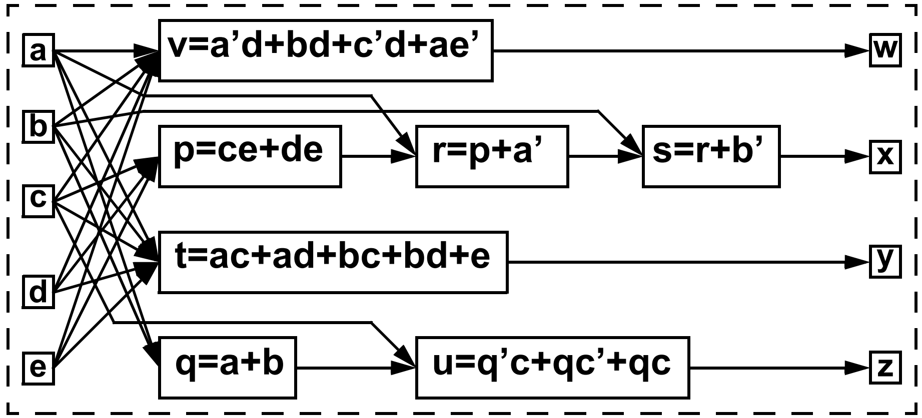
- Üks funktsioon *jagatakse väiksemateks* funktsioonideks
- Luuakse uus sõlm (uued sõlmed)
- Näide
  - *v* jagatakse kaheks
    `v=a'd+bd+c'd+ae';`
  - luuakse *j*
    `j=a'+b+c';   v=jd+ae';`



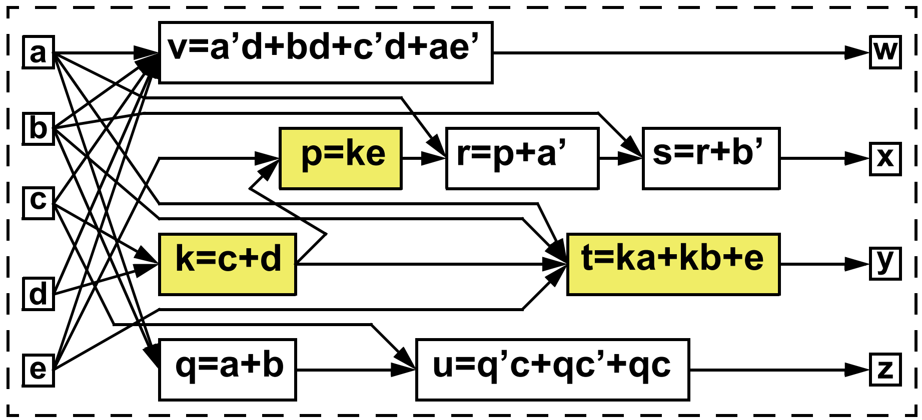


Eraldamine – Extraction

- Leitakse *ühine alam-avaldis* (common sub-expression) kahele (või enamale) funktsioonile
- Alam-avaldis eraldatakse kui uus funktsioon
- Luuakse uus sõlm



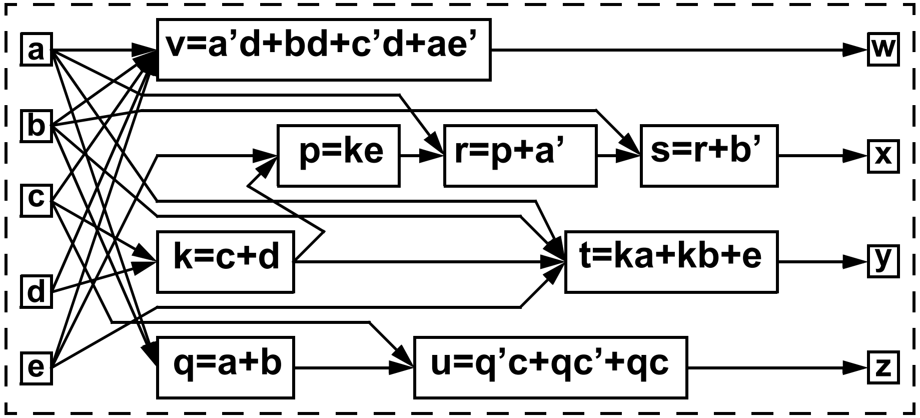
- Näide
  - *p* ja *t* jagavad alam-avaldist
    `p=ce+de;   t=ac+ad+bc+bd+e;`
    `p=(c+d)e;   t=(c+d)(a+b)+e;`
  - luuakse *k*
    `k=c+d;   p=ke;   t=ka+kb+e;`

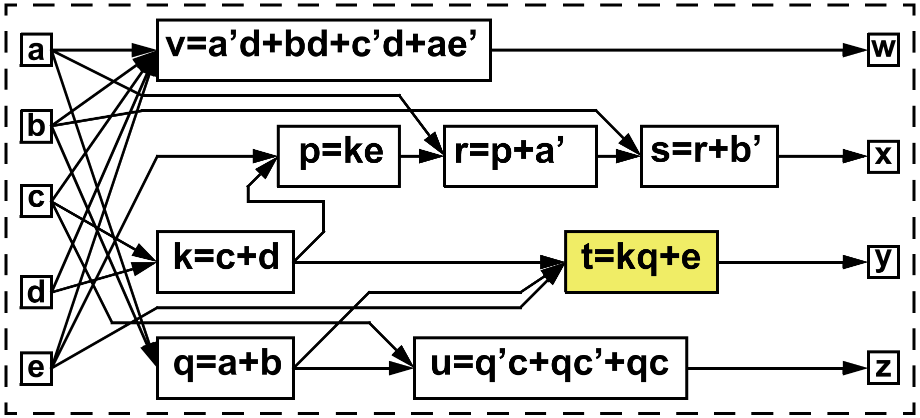


Asendamine – Substitution

- **Lihtsustatakse osa-funktsiooni kasutades *lisa-sisendit*, mida varem selles funktsioonis ei esinenud**

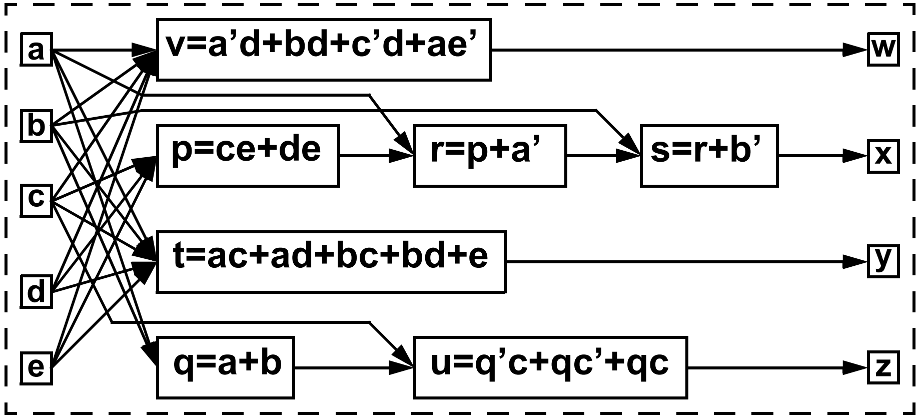
- **Näide**
  - ***t* sisaldab *q* kui alam-avaldist**
    `t=ka+kb+e;`
    `q=a+b;`
  - **uus *t***
    `t=kq+e;`

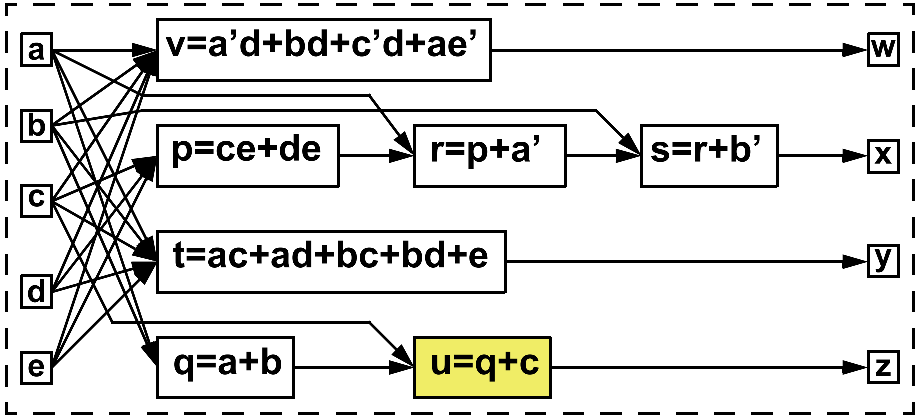




Lihtsustamine – Simplification

- *Lihtsustatakse* osa-funktsioon
- Näide
  - lihtsustatakse *u*
    u=q′c+qc′+qc
  - uus *u*
    u=q+c

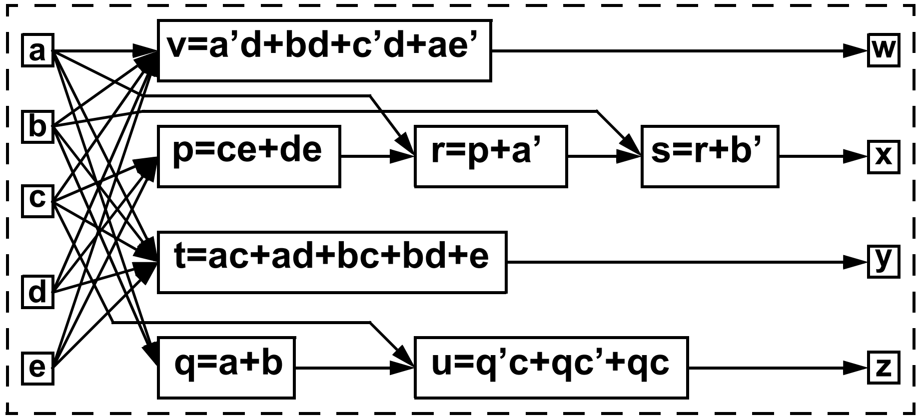




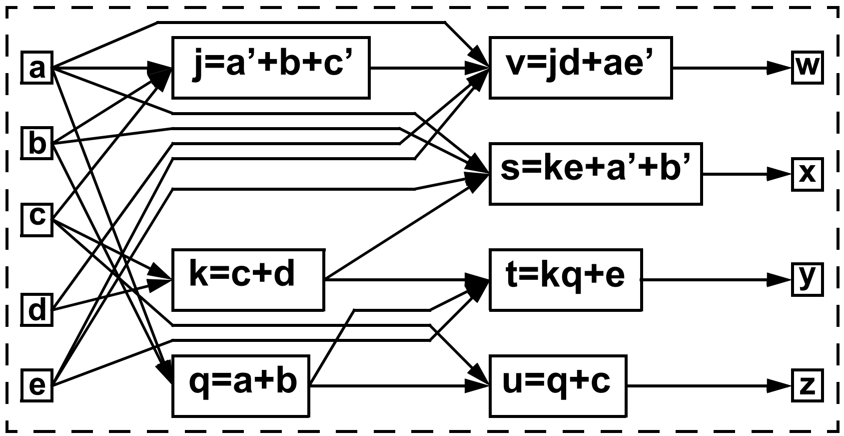
TTÜ1918

Teisenduste jada – lõplik tulemus

- **Lõplik funktsioon**
  ```
  j=a'+b+c'
  k=c+d
  q=a+b
  s=ke+a'+b'
  t=q+c
  u=q+c
  v=jd+ae'
  ```



- **Funktsioone/loogikalülisid – 7 ja 7**
- **Literaale – 33 ja 20**
- **Viide – 3 ja 2 (sõlmi)**
- **Viide – 9 ja 7 (sõlmi+literaale)**



Teisendused – kõik sammud

- **Lähteülesanne**
  `p=ce+de;   q=a+b;   r=p+a';   s=r+b';   t=ac+ad+bc+bd+e;`
  `u=q'c+qc'+qc;   v=a'd+bd+c'd+ae';   w=v;   x=s;   y=t;   z=u`
- **Eemaldamine – s, r → s**
  `p=ce+de;   q=a+b;   s=p+a'+b';   t=ac+ad+bc+bd+e;`
  `u=q'c+qc'+qc;   v=a'd+bd+c'd+ae';   w=v;   x=s;   y=t;   z=u`
- **Dekompositsioon – v → j, v**
  `p=ce+de;   q=a+b;   s=p+a'+b';   t=ac+ad+bc+bd+e;`
  `u=q'c+qc'+qc;   j=a'+b+c';   v=jd+ae';   w=v;   x=s;   y=t;   z=u`
- **Eraldamine – p, t → k, p, t**
  `j=a'+b+c';   k=c+d;   p=ke;   q=a+b;   s=p+a'+b';   t=ka+kb+e;`
  `u=q'c+qc'+qc;   v=jd+ae';   w=v;   x=s;   y=t;   z=u`
- **Asendamine – q, t → q, t**
  `j=a'+b+c';   k=c+d;   p=ke;   q=a+b;   s=p+a'+b';   t=kq+e;`
  `u=q'c+qc'+qc;   v=jd+ae';   w=v;   x=s;   y=t;   z=u`
- **Lihtsustamine – u → u**
  `j=a'+b+c';   k=c+d;   p=ke;   q=a+b;   s=p+a'+b';   t=kq+e;`
  `u=q+c;   v=jd+ae';   w=v;   x=s;   y=t;   z=u`
- **Eemaldamine – s, p → s**
  `j=a'+b+c';   k=c+d;   q=a+b;   s=ke+a'+b';   t=kq+e;`
  `u=q+c;   v=jd+ae';   w=v;   x=s;   y=t;   z=u`

Eemaldamine – näitealgoritm

- Defineeritakse läviväärtus k (tavaliselt 0)
- Kontrollitakse kõiki avaldisi (funktsioone)
- Avaldis eemaldatakse, kui literaalide arvu *kasv* ei ületa läviväärtust

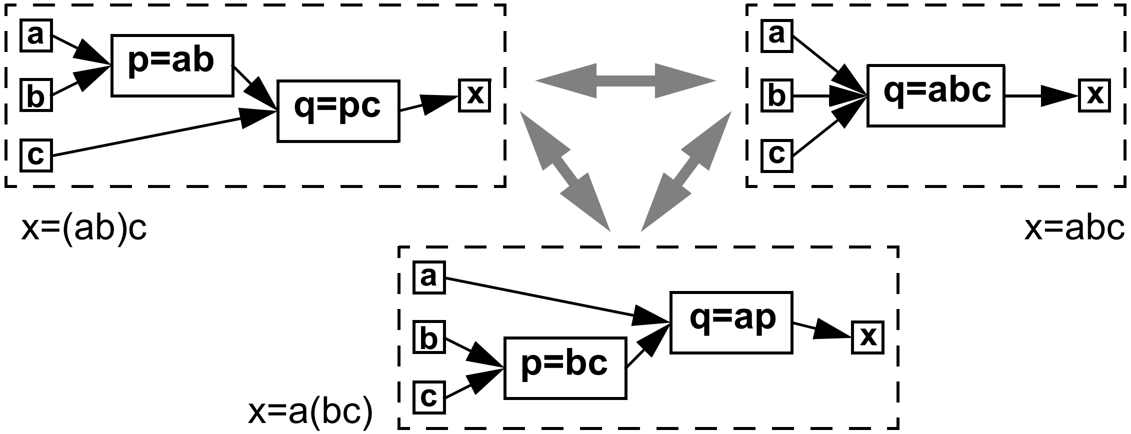
```
ELIMINATE ( Gn(V,E), k ) {
    repeat {
        vx = selected vertex with value < k ;
        if ( vx = ∅ ) return;
        replace x by fx in the network;
    }
}
```

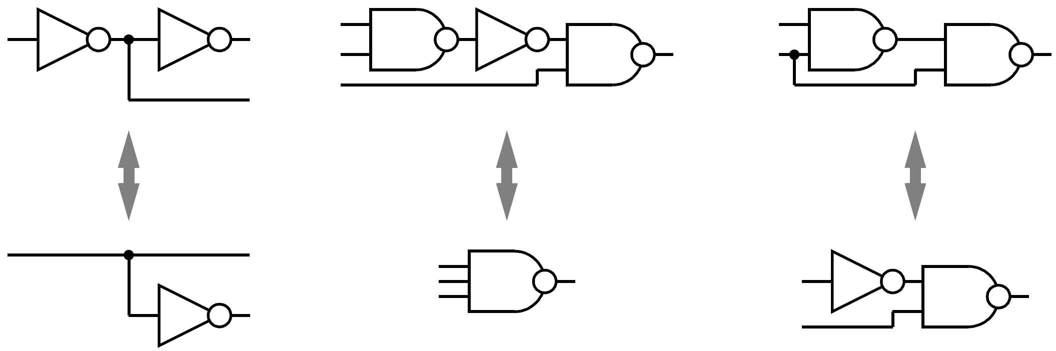
Optimeerimisviisid

- Algoritmiline
  - iga teisenduse tüübi jaoks tuleb määrata algoritm
  - algoritm moodustab *operaatori*
    - heuristilised meetodid
    - nõrk lokaalne optimeerimine
  - operaatorite järjekord
    - skriptipõhine
    - puhtkogemuslik
  - teisendused – algebralised ja kahendmeetodid
- Reeglitel põhinev
  - andmebaas – mustripaaride hulk
  - mustrite asendused defineeritud reeglite hulgaga
- Tehnoloogiast sõltuv optimeerimine (technology mapping)
  - elementide teegid – elemente realiseerib lihtsamat mõne sisendiga funktsiooni
  - programmeeritav loogika – mõne sisendiga suvalised funktsioonid (CLB)

Reeglitel põhinev optimeerimine

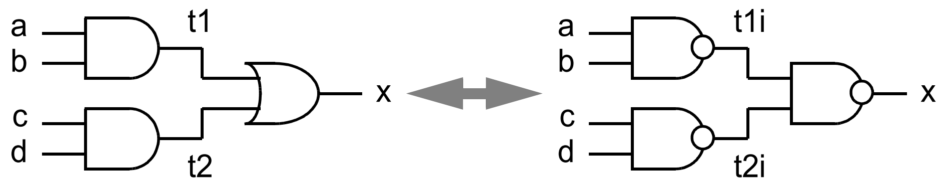
- **Mustri (pattern) otsimine ja asendamine teisega**
  - **vajadus kanoonilise esitusviisi järele**
- **Primitiivsete operatsioonide võrkgraaf**
  - **mustrite keerukus pole piiratud**
- **Seotus kasutatava tehnoloogiaga**
  - **abstrakne tehnoloogia**



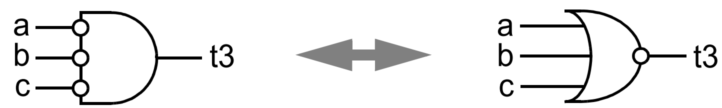


Reeglitel põhinevad teisendused

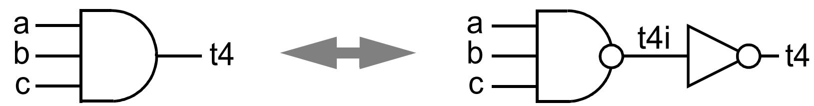
- **De Morgani seadus**
  - **t1 = a b ;   t2 = c d ;   x = t1 + t2 ;**
  - **t1’ = (a b)’ ;   t2’ = (c d)’ ;   x = (t1’ t2’)’ ;**
  - **t1i = (a b)’ ;   t2i = (c d)’ ;   x = (t1i t2i)’ ;**



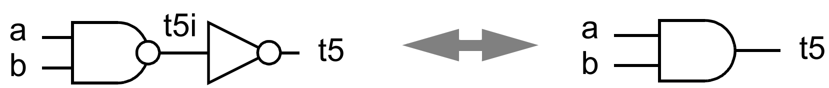
  - **t3 = a’ b’ c’ ;**
  - **t3 = (a + b + c)’ ;**



  - **t4 = a b c ;**
  - **t4i = (a b c)’ ; t4 = t4i’ ;**



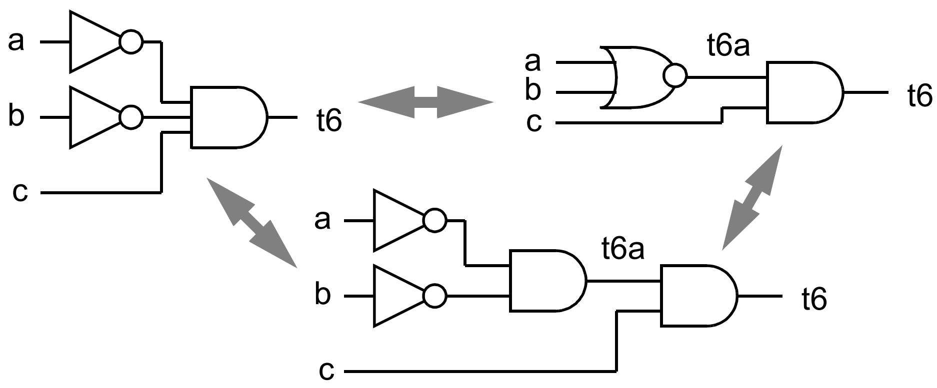
- **Topelteituse seadus**
  - **t5i = (a b)’ ;   t5 = t5’ ;**
  - **t5 = a b ;**



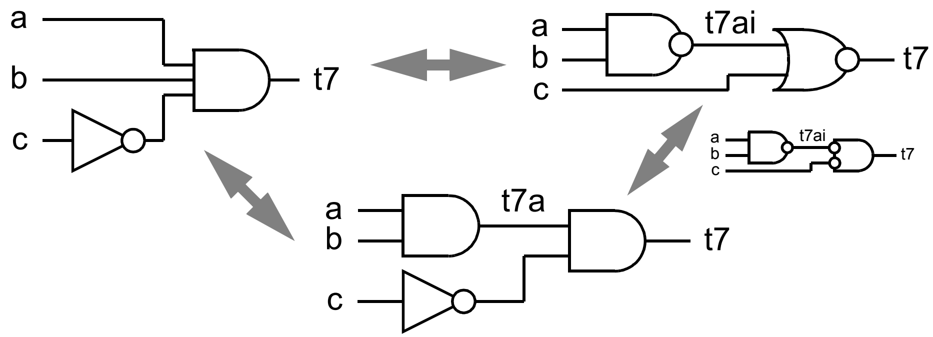
Reeglitel põhinevad teisendused

- **De Morgani & topelteituse seadused**

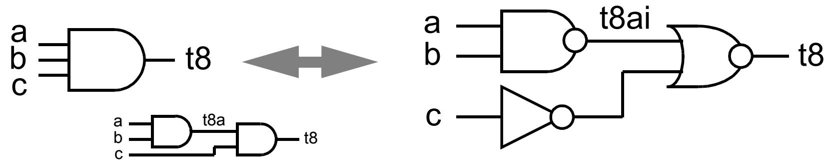
  - **t6 = a' b' c ;**
  - **t6a = a' b' ; t6 = t6a c ;**
  - **t6a = (a + b)' ; t6 = t6a c ;**



  - **t7 = a b c' ;**
  - **t7a = a b ; t7 = t7a c' ;**
  - **[t7ai = (a b)' ; t7 = t7ai' c' ;]**
  - **t7ai = (a b)' ; t7 = (t7ai + c)' ;**



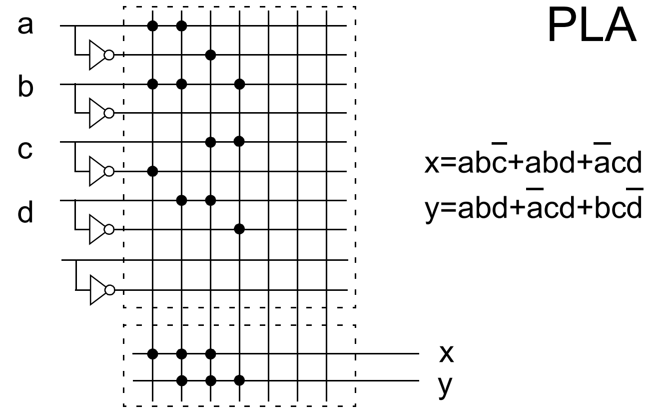
  - **t8 = a b c ;**
  - **[t8a = a b ; t8 = t8a c ;]**
  - **t8ai = (a b)' ; t8 = (t8ai + c')' ;**



Erijuhud – programmeeritav loogika

PLD – Programmable Logic Device

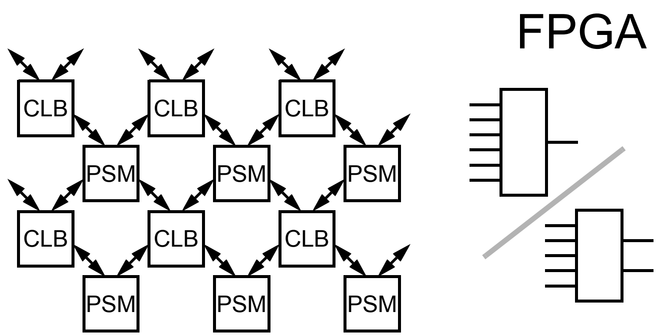
- **PLA – Programmable Logic Array**
  - **Programmeeritavad loogikamaatriksid (ja ühendused)**
  - **Loogikamaatriks**
    - **mitme sisendiga ja mitme väljundiga loogika-funktsioonide süsteem, implikantide arv piiratud**
  - **Tükeldamine vastava suurusega funktsioonideks**



$x=ab\bar{c}+abd+\bar{a}cd$

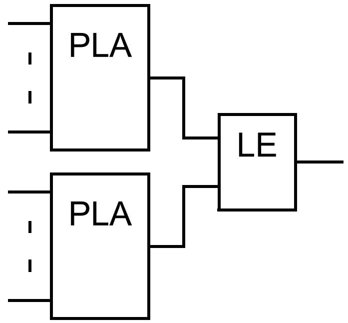
$y=abd+\bar{a}cd+bc\bar{d}$

- **FPGA – Field Programmable Gate Array**
    - **väliprogrammeeritav loogika (ka korduvprogr. loogika)**
  - **Programmeeritavad loogikaplokid (CLB)**
    - **... ja programmeeritavad ühendused (PSM)**
  - **Xilinx – Spartan, Artix, Kintex & Virtex seeriad**
    - **CLB – neli 6->1 / 5->2 funktsiooni**
  - **Tükeldamine 5- või 6-sisendiga funktsioonideks**



Erijuhud – 3-tasemeline realisatsioon

- **Kombinatsioon FPGA-st ja PLA-st**
  - **Kahe PLA väljundid on ühendatud läbi loogika-elemendi**
    - **AND-OR-XOR ~~ summaatorid / korrutid**
    - **AND-OR-AND ~~ kahe funktsiooni ühisosa**
    - **AND-OR-OR ~~ kaks funktsiooni paralleelselt (1-d)**
    - **OR-AND-OR ~~ kaks funktsiooni paralleelselt (0-d)**



cd \ ab	00	01	11	10
00	0	1	0	1
01	1	1	1	0
11	0	1	0	1
10	1	0	1	1

AND-OR - 8 impl., 24 lit.:

$f = \bar{a}b\bar{c} + \bar{a}bd + a\bar{b}\bar{d} + a\bar{b}c +$
$+ \bar{a}\bar{c}d + b\bar{c}d + \bar{b}c\bar{d} + ac\bar{d}$

AND-OR-XOR - 4 impl., 8 lit.:

$f = (\bar{a}b + \bar{c}d) \oplus (a\bar{b} + c\bar{d})$

de \ abc	000	001	011	010	110	111	101	100
00	1	0	0	0	1	0	0	0
01	0	1	0	0	0	1	0	0
11	0	0	1	0	0	0	1	0
10	0	0	0	1	0	0	0	1

AND-OR - 8 impl., 40 lit.:

$f = \bar{a}\bar{b}\bar{c}\bar{d}\bar{e} + \bar{a}\bar{b}c\bar{d}e + \bar{a}bcde + ab\bar{c}d\bar{e} +$
$+ ab\bar{c}\bar{d}\bar{e} + abc\bar{d}e + a\bar{b}cde + a\bar{b}\bar{c}d\bar{e}$

AND-OR-AND - 6 impl., 16 lit.:

$f = (\bar{a}\bar{b}\bar{d} + \bar{a}bd + ab\bar{d} + a\bar{b}d) \cdot (\bar{c}\bar{e} + ce)$

Funktsioonide teisendused – kahendmeetodid

- **Loogikafunktsioonide omaduste kasutamine**
- **Võimalik kasutada osaliselt määratust (don't-care)**
- **Aegajalt (liigagi) keeruline**

- **Kahendasendus (boolean substitution)**
  - **lähtefunktsioonid**
    `h=a+bcd+e;  q=a+cd`
  - **tulemus**
    `h=a+bq+e`
  - **sest**
    `a+bq+e=a+b(a+cd)+e=a+bcd+e` `[a+b(a+cd)+e=a+ab+bcd+e=a+bcd+e]`
  - **või hoopis?!**
    `h=a+bcd+e;  q=cd+e  ->  h=a+bq+e`
    `a+bq+e=a+b(cd+e)+e=a+bcd+e` `[a+b(cd+e)+e=a+bcd+be+e=a+bcd+e]`

Funktsioonide teisendused – algebralise meetodid

- Funktsioone vaadeldakse kui *polünoome*
- Kasutatakse polünoomide algebra omadusi
- Lihtsam, kiirem, kui nõrgem optimeerimisvõime

- Algebraline asendus (algebraic substitution)
  - lähtefunktsioon
    `t=ka+kb+e`
  - tulemus
    `t=kq+e`
  - sest
    `q=a+b`

Algebralised meetodid

- *Polünoom* (polynomial) – korrutiste summa
- *Monoom* (monomial), üksliige – üksik korrutis e. kuup
- Ainult distributiivsuse seadus kasutusel
  - $a(b+c)=ab+ac$, kuid $a+bc \neq (a+b)(a+c)$
- Täiendid pole defineeritud
  - muutuja täiendit vaadeldakse kui lisa muutujat
- Määramatused pole defineeritud
- Operatsioonid ainult avaldistega, mille muutujate hulgad ei kattu
- Liiaste kuupide eemaldamine pole võimalik
  - $(a+b)$ ja $(c+d) \rightarrow (a+b)(c+d)=ac+ad+bc+bd$ OK!
  - $(a+b)$ ja $(a+c) \rightarrow (a+b)(a+c)=aa+ac+ba+bc \neq a+bc$ ei
  - $(a+b)$ ja $(\bar{a}+c) \rightarrow (a+b)(\bar{a}+c)=a\bar{a}+ac+b\bar{a}+bc \neq ac+b\bar{a}$ ei

Algebraline jagatis

- Kaks algebralist avaldist
- *jagatav* (dividend), *jagaja* (divisor), *jagatis* (quotient), *jääk* (remainder)
- $f_{jagatis} = f_{jagatav} / f_{jagaja}$, kui
- $f_{jagatav} = f_{jagaja} \cdot f_{jagatis} + f_{jääk}$
- $f_{jagaja} \cdot f_{jagatis} \neq \varnothing$
- ning f_{jagaja} ja $f_{jagatis}$ muutujate hulgad ei kattu ($sup(f_{jagaja}) \cap sup(f_{jagatis}) = \varnothing$)
- $f_{jagatav} = ac + ad + bc + bd + e$ & $f_{jagaja} = a + b$
- $f_{jagatis} = c + d$ & $f_{jääk} = e$
- $(a+b)(c+d)+e=f_{jagatav}$ & $\{a,b\} \cap \{c,d\} = \varnothing$
- Mitte-algebraline jagatis – $f_i = a + bc$ & $f_j = a + b$
  - $(a+b)(a+c)=f_i$ kuid $\{a,b\} \cap \{a,c\} \neq \varnothing$

Jagamisalgoritm

- $A = \{ C_j^A, j=1,2,...,l\}$ – jagatava kuupide hulk
- $B = \{ C_i^B, i=1,2,...,n\}$ – jagaja kuupide hulk
- Jagatis Q ja jääk R on kuupide summad

```
ALGEBRAIC_DIVISION (A,B) {
    for (i=1 to n) {
        D={Cj^A such that Cj^A ⊇ Ci^B};
        if (D==∅) return(∅,A);
        Di=D with var. in sup(Ci^B) dropped;
        if (i==1)   Q=Di;   else Q=Q∩Di;
    }
    R=A-Q×B;
    return(Q,R);
}
```

Jagamine – näide #1

- $f_{jagatav}$ = ac + ad + bc + bd + e & f_{jagaja} = a + b
- A = {ac, ad, bc, bd, e} & B = {a, b}
- *i = 1*
  - C_1^B=a, D={ac, ad} & D_1={c,d}
  - Q = {c,d}
- *i = 2*
  - C_2^B=b, D={bc,bd} & D_2={c,d}
  - Q = {c,d}$\cap${c,d} = {c,d} -- *kuup vastab elemendile!*
- Tulemus
  - Q = {c,d} & R = {e}
  - $f_{jagatis}$=c+d & $f_{jääk}$=e

Jagamine – näide #2

- $f_{jagatav} = axc + axd + bc + bxd + e$ & $f_{jagaja} = ax + b$
- A = {axc, axd, bc, bxd, e} & B = {ax, b}
- *i = 1*
  - C_1^B=ax, D={axc, axd} & D_1={c,d}
  - Q = {c,d}
- *i = 2*
  - C_2^B=b, D={bc,bxd} & D_2={c,xd}
  - Q = {c,d}∩{c,xd} = {c} -- *kuup vastab elemendile!*
- Tulemus
  - Q = {c} & R = {axd,bxd,e}
  - $f_{jagatis}$=c & $f_{jääk}$=axd+bxd+e

Jagamine – mis siis ikkagi toimub?

- A = ac + ad + bc + bd + e & B = a + b
  - (1) $\underline{a}$ ($\underline{c+d}$) + bc + bd + e
  - (2) a (c+d) + $\underline{b}$ ($\underline{c+d}$) + e → (c+d) "∩" (c+d) "=" (c+d)
  - (R) [ac+ad+bc+bd+e] "-" [(a+b) (c+d)] "="
    "=" [ac+ad+bc+bd+e] "-" [ac+ad+bc+bd] "=" [e]
  - ac+ad+bc+bd+e = a($\underline{c+d}$)+b($\underline{c+d}$)+e = (c+d)(a+b)+e

- A = axc + axd + bc + bxd + e & B = ax + b
  - (1) $\underline{ax}$ ($\underline{c+d}$) + bc + bxd + e
  - (2) ax (c+d) + $\underline{b}$ ($\underline{c+xd}$) + e → (c+d) "∩" (c+xd) "=" (c)
  - (R) [axc+axd+bc+bxd+e] "-" [(ax+b) (c)] "="
    "=" [axc+axd+bc+bxd+e] "-" [axc+bc] "=" [axd+bxd+e]
  - axc+axd+bc+bxd+e= ax($\underline{c}$+d)+b($\underline{c}$+xd)+e = ax($\underline{c}$)+b($\underline{c}$)+axd+bxd+e = c(ax+b)+axd+bxd+e

Jagatise eksisteerimine?

- Teoreem
- Antud kaks algebralist avaldist f_i ja f_j
- f_i / f_j on tühi, kui üks järgnevaist tingimustest on täidetud:
  - f_j sisaldab muutujat, mida pole f_i-s;
  - f_j sisaldab kuupi, mille tugimuutujad ei sisaldu üheski f_i kuubi tugimuutujate hulgas ($\exists$ $sup(C^j) \not\subset sup(C^i)$, $\forall C^i \in f_i$);
  - f_j sisaldab rohkem liikemid kui f_i;
  - suvalist muutujat on f_j-s rohkem kui f_i-s.
- Kasutusel kiireks kontrolliks
  - jagatist ei pruugi ikkagi eksisteerida – ac + be / a + b

Jagatise eksisteerimine?

- Antud kaks algebralist avaldist f_i ja f_j
- f_i / f_j on tühi, kui üks järgnevaist tingimustest on täidetud:
  - f_j sisaldab muutujat, mida pole f_i-s;

    ab + cd / a + e a(b) + e(?) + cd
  - f_j sisaldab kuupi, mille tugimuutujad ei sisaldu üheski f_i kuubi tugimuutujate hulgas ($\exists$ sup(C^j)$\not\subset$sup(C^i), $\forall C^i \in f_i$);

    abc + def / ab + ad ab(c) + ad(?) + def
  - f_j sisaldab rohkem liikemid kui f_i;

    ab + cd / a + b + c a(b) + b(?) + c(d)
  - suvalist muutujat on f_j-s rohkem kui f_i-s.

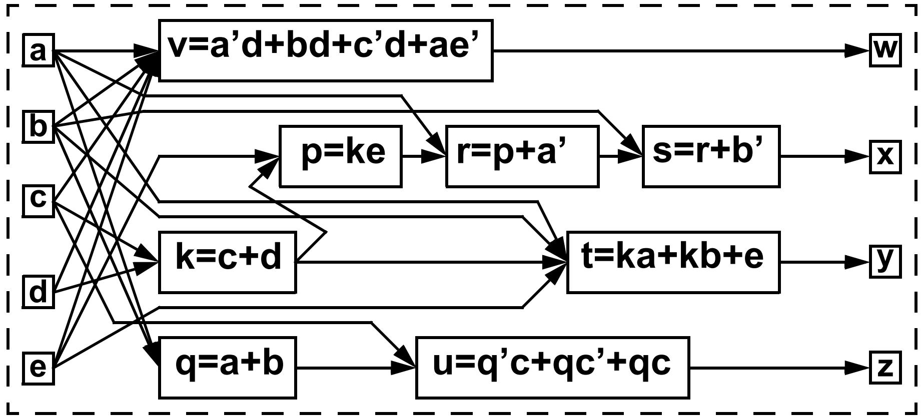
    abc + ade + bcd / ab + ad + ac ab(c) + ad(e) + ac(?)

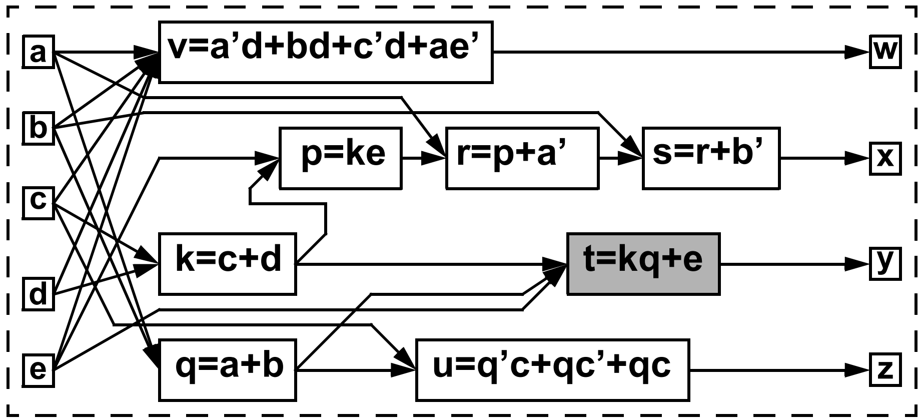
- Osaline kattumine? ab + ac + bc / a + b = ? ab + ac + bc = ab + (a+b) c

Asendamine

- Vaadeldakse avaldiste paare
- Jagamine suvalises järjekorras
- Kui jagatis pole tühi, siis:
  - ennustatakse pindala/viite muutust
  - $f_{jagatav}$ asendatakse $j \cdot f_{jagatis} + f_{jääk}$'ga, kus $j = f_{jagaja}$

  - $f_t = ka+kb+e$ & $f_q = a+b$
  - $f_{jagatis} = k$ & $f_{jääk} = e$
  - $f_t = kq+e$





Eraldamine

- **Ühiste alam-avaldiste otsimine**
  - **üksikute kuupide eraldamine – monoom**
  - **mitme kuubi eraldamine – *tuum* (kernel)**
- **Sobivate *jagajate* leidmine**

- ***Kuubivaba* (cube-free) avaldis**
  - **pole võimalik faktoriseerida kuupi kasutades**
    - **ehk siis, mitte ühtegi muutujat ei saa sulgude ette tuua**
- **Avaldise *tuum***
  - **avaldise kuubivaba jagatis, kui jagaja on kuup (*kaas-tuum* (co-kernel))**
- **Avaldise *tuumade hulk* K(f)**

Tuumad – näide

- f_x = ace + bce + de + g
  - f_x / a → ce → ei ole kuubivaba
  - f_x / b → ce → ei ole kuubivaba
  - f_x / c → ae + be → ei ole kuubivaba
  - f_x / ce → a + b → kuubivaba → *tuum*
  - f_x / d → e → ei ole kuubivaba
  - f_x / e → ac+bc+d → kuubivaba → *tuum*
  - f_x / g → 1 → ei ole kuubivaba
  - f_x / 1 → ace+bce+de+g → kuubivaba → *tuum*
- K(fx) = { (a+b), (ac+bc+d), (ace+bce+de+g) }

Kahe avaldise ühine mitmik-kuup jagaja
Teoreem (Brayton & McMullen)

- **Kahel avaldisel f_a ja f_b leidub ühine mitmik-kuup jagaja f_d siis ja ainult siis, kui leiduvad sellised tuumad $k_a \in K(f_a)$ ja $k_b \in K(f_b)$, et f_d on 2 (või enama) kuubi summa k_a ja k_b ühisosas ($|k_a \cap k_b| \geq 2$)**

- **Järeldus**
  ühise alam-avaldise **otsimise võib ära jätta, kui tuumade hulkade ühisosa on** ***tühihulk***

Eraldamine – näide

- f_x = ace + bce + de + g
- f_y = ad + bd + cde + ge
- f_z = abc

- $K(f_x)$ = { (a+b), (ac+bc+d), (ace+bce+de+g) }
- $K(f_y)$ = { (a+b+ce), (cd+g), (ad+bd+cde+ge) }
- $K(f_z)$ = {}

- f_w = a+b
- f_x = wce + de + g
- f_y = wd + cde + ge
- f_z = abc

Tuumade hulga leidmine

- Naiivne lähenemine
  - üritatakse leida jagatised muutujate kombinatsioonidele
  - eemaldatakse mitte-kuubivabad jagatised
- Kaval lähenemine
  - kasutatakse rekursiooni – tuumade tuumad on samuti tuumad
  - kasutatakse korrutamise kommutatiivsus-omadust
    - väldib liigseid arvutusi
      jagada a-ga ja siis b-ga
      jagada b-ga ja siis a-ga

Dekompositsioon

- $f_x = ace + bce + de + g;$
- $f_t = ac + bc + d;$ $f_x = te + g;$
- $f_s = a + b;$ $f_t = sc + d;$ $f_x = te + g;$

- **Tuumadel põhinev dekompositsioon**
  - avaldist jagatakse rekursiivselt
  - $f_x = ace + bce + de + g$
  - $K(f_x) = \{ (a+b), \underline{(ac+bc+d)}, (ace+bce+de+g) \}$
  - $f_t = ac + bc + d;$ $f_x = te + g;$
  - $K(f_t) = \{ \underline{(a+b)}, (ac+bc+d) \}$
  - $f_s = a + b;$ $f_t = sc + d;$ $f_x = te + g;$

Näide – minimeerimine 1-de järgi

ülesanne	
abcd	klmn
0000	-00-
0001	1--0
0010	-0-1
0011	111-
0100	0-1-
0101	-1--
0110	0011
0111	1-01
1000	1000
1001	11-1
1010	-000
1011	0000
1100	0111
1101	-111
1110	11-1
1111	1-11

espresso		
abcd	klmn	
0-10	0011	1
-00-	1000	2
1-01	0101	3
00-1	1110	4
-1-1	1001	5
111-	1111	6
-10-	0111	7

4 NOT
3 2-AND
4 3-AND
3 4-OR
1 5-OR

4 + 18 + 17

k

		d	d, c	c
	-	1	1	-
b	0	-	1	0
b, a	0	-	1	1
a	1	1	0	-

l

		d	d, c	c
	0	-	1	0
b	-	1	-	0
b, a	1	1	-	1
a	0	1	0	0

m

		d	d, c	c
	0	-	1	-
b	1	-	0	1
b, a	1	1	1	-
a	0	-	0	0

n

		d	d, c	c
	-	0	-	1
b	-	-	1	1
b, a	1	1	1	1
a	0	1	0	0

TTÜ1918

Näide – minimeerimine 1-de järgi – tuumad

tulemus

abcd	klmn
0-10	0011
-00-	1000
1-01	0101
00-1	1110
-1-1	1001
111-	1111
-10-	0111

4 NOT
3 2-AND
4 3-AND
3 4-OR !!
1 5-OR !!

4+18+17=39

$k = \bar{b}\,\bar{c} + \bar{a}\,\bar{b}\,\bar{d} + b\,d + a\,b\,c$

$l = a\,\bar{c}\,d + \bar{a}\,\bar{b}\,d + a\,b\,c + b\,\bar{c}$

$m = \bar{a}\,c\,\bar{d} + \bar{a}\,\bar{b}\,d + a\,b\,c + b\,\bar{c}$

$n = \bar{a}\,c\,\bar{d} + a\,\bar{c}\,d + b\,d + a\,b\,c + b\,\bar{c}$

legaalsed loogikaelemendid

$k1 = \bar{a}\,\bar{b}\,d + a\,b\,c$

$k = k1 + \bar{b}\,\bar{c} + b\,d$

$l = k1 + a\,\bar{c}\,d + b\,\bar{c}$

$m1 = \bar{a}\,c\,\bar{d} + b\,\bar{c}$

$m = k1 + m1$

$n1 = a\,\bar{c}\,d + b\,d + a\,b\,c$

$n = m1 + n1$

4 NOT, 3*2-AND, 4*3-AND, 4*2-OR, 3*3-OR
literaale: 4+18+17=39
eq.gates: 4*1.5+3*2.0+4*2.5+4*2.0+3*2.5=37.5

tuumad

k:	$/\bar{b}$:	$\bar{c} + \bar{a}\,d$		
	$/b$:	$d + a\,c$		[1]
	$/d$:	$\bar{a}\,\bar{b} + b$	$(=\bar{a}+b)$	
l:	$/a$:	$\bar{c}\,d + b\,c$		[2]
	$/b$:	$a\,c + \bar{c}$	$(=a+\bar{c})$	[3]
	$/\bar{c}$:	$a\,d + b$		[4]
	$/d$:	$a\,\bar{c} + \bar{a}\,\bar{b}$		
m:	$/\bar{a}$:	$c\,\bar{d} + \bar{b}\,d$		
	$/b$:	$a\,c + \bar{c}$	$(=a+\bar{c})$	[3]
	$/c$:	$\bar{a}\,\bar{d} + a\,b$		[5]
n:	$/a$:	$\bar{c}\,d + b\,c$		[2]
	$/b$:	$d + a\,c + \bar{c}$	$(=d+a+\bar{c})$	[1&3]
	$/c$:	$\bar{a}\,\bar{d} + a\,b$		[5]
	$/\bar{c}$:	$a\,d + b$		[4]
	$/d$:	$a\,\bar{c} + b$		

Näide – minimeerimine 1-de järgi – jagamine

$k = \bar{b}\,\bar{c} + \bar{a}\,\bar{b}\,\bar{d} + b\,d + a\,b\,c$

$l = a\,\bar{c}\,d + \bar{a}\,\bar{b}\,d + a\,b\,c + b\,\bar{c}$

$m = \bar{a}\,c\,\bar{d} + \bar{a}\,\bar{b}\,d + a\,b\,c + b\,\bar{c}$

$n = \bar{a}\,c\,\bar{d} + a\,\bar{c}\,d + b\,d + a\,b\,c + b\,\bar{c}$

ühised tuumad

[1] $(d + a\,c)$ k,n;   [3] $(a\,c + \bar{c})$ l,m

[2] $(\bar{c}\,d + b\,c)$ l,n

[3] $(a\,c + \bar{c})$ l,m,n

[4] $(a\,d + b)$ l,n

[5] $(\bar{a}\,\bar{d} + a\,b)$ m,n

	NOT	2-AND	3-AND	2-OR	3-OR	lit.	e.g.
-	4	3	4	4	3	39	37.5
1&3	4	5	3	3	4	41	39.5
2	4	7	4	4	3	47	45.5
3	4	3	4	3	4	40	38.0
4	4	5	3	4	3	40	39.0
5	4	7	2	5	2	40	40.0

[3] $(a\,c + \bar{c})$ l,m,n

$t1 = a\,c + \bar{c}$ $[= a + \bar{c}]$

[l/t1: $l = a\,c\,(\underline{b}) + \bar{c}\,(a\,d + \underline{b}) + \bar{a}\,\bar{b}\,d$]

[m/t1: $m = a\,c\,(\underline{b}) + \bar{c}\,(\underline{b}) + \bar{a}\,c\,\bar{d} + \bar{a}\,\bar{b}\,d$]

[n/t1: $n = a\,c\,(\underline{b}) + \bar{c}\,(\underline{b}) + \bar{a}\,c\,\bar{d} + a\,\bar{c}\,d + b\,d$]

$k1 = \bar{b}\,\bar{c} + b\,d$

$k = k1 + \bar{a}\,\bar{b}\,d + a\,b\,c$

$l = t1\,b + \bar{a}\,\bar{c}\,d + \bar{a}\,\bar{b}\,d$

$m = t1\,b + \bar{a}\,c\,\bar{d} + \bar{a}\,\bar{b}\,d$

$n1 = t1\,b + b\,d$

$n = n1 + \bar{a}\,c\,\bar{d} + a\,\bar{c}\,d$

4*NOT, 3*2-AND, 4*3-AND, 3*2-OR, 4*3-OR
literaale: 4+18+18=40
eq.gates: 4*1.5+3*2.0+4*2.5+3*2.0+4*2.5=38.0

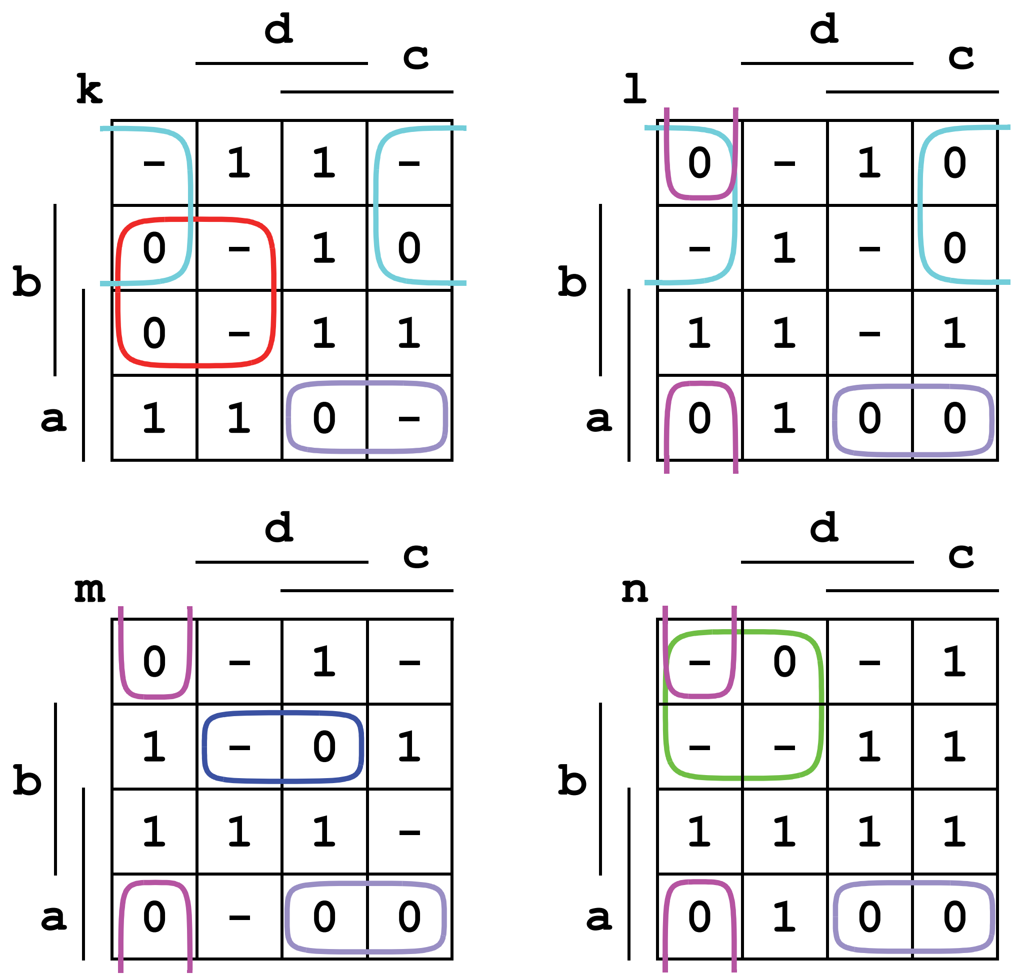
Näide – minimeerimine 1-de ja 0-de järgi

espresso -Dopoall

```
     0000
abcd klmn
01-1 0010 1
0-0- 0001 2
-10- 1000 3
0--0 1100 4
-000 0111 5
101- 1111 6
```

8 NOT
3 2-OR
3 3-OR
4 3-AND

8+15+12=35
eq.gates: 35.5



Ainult NOR elemendid!

$k = ((\overline{b}+c)'+(a+d)'+(\overline{a}+b+\overline{c})')'$

$l = ((a+d)'+(b+c+d)'+(\overline{a}+b+\overline{c})')'$

$m = ((a+\overline{b}+\overline{d})'+(b+c+d)'+(\overline{a}+b+\overline{c})')'$

$n = ((a+c)'+(b+c+d)'+(\overline{a}+b+\overline{c})')'$

4*NOT, 3*2-NOR, 7*3-NOR
literaale: 4+27=31
eq.gates: 4*1.5+3*1.5+7*2.0=24.5