2 skyrius
Teiginių kalba

2.1 Loginės operacijos

Kai kurios sąvokos (aibė, taškas,...) matematikoje yra *pirminės.* Jos neapibrėžiamos, o paaiškinamos. Tokia sąvoka logikoje yra *teiginys.*

Teiginiu vadinsime sakinį, kurio atžvilgiu prasmingas klausimas "Teisingas jis ar klaidingas?"

Pavyzdžiui, *Jonukas gerai mokosi, Vilnius yra Lietuvos sostinė, sausis šalčiausias metų mėnuo.*

Teiginius žymėsime raidėmis p, q, r, s bei su indeksais $p_0, q_0, r_0, s_0, p_1, q_1, r_1, s_1, ...$
Ne visi sakiniai yra teiginiai. Pavyzdžiui, *Kelinta valanda? Šlovė mokslui!* Ne visada galima pasakyti teisingas ar klaidingas duotasis teiginys. Tai gali būti neišspręstų problemų formulavimai arba, pavyzdžiui, nežinomi tvirtinimai apie praeitį bei ateitį.

Jei teiginys p teisingas, tai sakysime, kad teiginio p **vertė** lygi t ir žymėsime $p = t$. Jei teiginys klaidingas, tai sakysime, kad teiginio **vertė** lygi k ir žymėsime $p = k$.

Apibrėšime šias logines operacijas: neigimą, konjunkciją, disjunkciją, griežtąją disjunkciją, implikaciją, ekvivalentumą, o vėliau ir Sheffer funkciją.
Pritaikę logines operacijas teiginiams, gausime sudėtinius teiginius.

Neigimas žymimas simboliu $\neg$ (pavyzdžiui, $\neg p, \neg q$). Literatūroje dar naudojami simboliai $^-$ ($\bar{p}, \bar{q}, ...$) bei $\sim$ ($\sim p, \sim q, ...$). $\neg p$ perskaitomas: "ne p", "netiesa, kad p". Sudėtinio teiginio $\neg p$ vertė priešinga teiginio p vertei: jei teiginio p vertė k, tai teiginio $\neg p$ vertė t, o jei teiginio p vertė k, tai teiginio $\neg p$ vertė t.

Visa tai patogu užrašyti lentele:

p	$\neg p$
t	k
k	t

Konjunkcija žymima simboliu & (dar naudojami $\wedge, \cdot$). $p\&q$ perskaitomas *p ir q*. Konjunkcija nusakoma lentele:

p,	q	$p\&q$
t	t	t
t	k	k
k	t	k
k	k	k

Sudėtinis teiginys $p\&q$ teisingas tada ir tiktai tada, kai abu teiginiai *p, q* teisingi.

Disjunkcija žymima simboliu $\vee$. $p \vee q$ perskaitomas *p arba q*. Disjunkcija nusakoma lentele:

p,	q	$p \vee q$
t	t	t
t	k	t
k	t	t
k	k	k

Sudėtinis teiginys $p \vee q$ klaidingas tada ir tiktai tada, kai abu teiginiai *p, q* klaidingi.

Griežtoji disjunkcija žymima simboliu $\dot{\vee}$ ($+$, $\oplus$). $p \dot{\vee} q$ perskaitomas, *arba p, arba q* . Griežtoji disjunkcija nusakoma lentele:

p,	q	$\dot{\vee}$
t	t	k
t	k	t
k	t	t
k	k	k

Sudėtinis teiginys $p \dot{\vee} q$ teisingas tada ir tiktai tada, kai teisingas lygiai vienas iš teiginių p ir q.

Implikacija žymima $\rightarrow$ ($\supset$, $\Rightarrow$). $p \rightarrow q$ perskaitomas *jei p, tai q* arba *iš p išplaukia q*. Terminas *iš p seka q* (autoriaus nuomone) irgi vartotinas, nes, kaip matysime vėliau, logikoje tokio sakinio teisingumas dažnai siejamas su **seka** formulių, tenkinančių tam tikras sąlygas.
Implikacija nusakoma lentele:

p,	q	$p \rightarrow q$
t	t	t
t	k	k
k	t	t
k	k	t

Sudėtinis teiginys $p \rightarrow q$ klaidingas tada ir tiktai tada, kai teiginys p teisingas, o teiginys q klaidingas. Pasistengsime pateisinti tokią teisingumo lentelę. Matematikoje aptinkami sakiniai pavidalu *jei p, tai q* laikomi teisingais, jei kiekvieną kartą kai p teisingas, q irgi yra teisingas. Atveju, kai p klaidingas, q gali įgyti bet kurią vertę. Pavyzdžiui, *jei* $x < y$ *ir* $y < z$, *tai* $x < z$ laikomas teisingu bet kurioms x, y, z reikšmėms. Galima parinkti tokius natūraliuosius x, y, z, kad "$x < y$ ir $y < z$" (t.y. p) būtų klaidingas, o "$x < z$" (t.y. q) vienu atveju būtų teisingas, kitu klaidingas, bet nėra tokių x, y, z, kad p būtų teisingas, o q klaidingas.

Ekvivalentumas žymimas $\leftrightarrow$ ($\Leftrightarrow$). $p \leftrightarrow q$ perskaitomas *p ekvivalentus q, p ir q ekvivalentūs.*
Ekvivalentumas nusakomas lentele:

p,	q	$p \leftrightarrow q$
t	t	t
t	k	k
k	t	k
k	k	t

Kaip matome į teiginius mes žiūrime kaip į kintamuosius, kurių kitimo sritis yra aibė $\{t, k\}$. Tokie kintamieji vadinami *propoziciniais (teigininiais) kintamaisiais.* Mes juos vadinsime taip, kaip įprasta vadinti informatikoje tokio tipo kintamuosius $-$ **loginiais kintamaisiais**. Pakeisime sudėtinių teiginių sąvoka tikslesne $-$ formulės sąvoka, o vietoje *vertė*, kaip ir įprasta kintamųjų atveju, vartosime sąvoką *reikšmė.* Kai kada bus patogu loginių kintamųjų kitimo sritimi laikyti aibę $\{1, 0\}$, t.y. t keisime 1, o $k - 0$. Dažniausiai mums net nesvarbu kokia tai aibė, o naudojames tik tuo, kad ji susideda iš dviejų elementų.

Formules žymėsime didžiosiomis lotyniškomis raidėmis A,B,C,...,F,..., kai kada su indeksais ar štrichais.

2.1 apibrėžimas.
1. *Loginis kintamasis yra formulė.*
2. *Jei* F *– formulė, tai* $\neg F$ *irgi formulė.*
3. *Jei* F, G *– formulės, tai* $(F \& G)$, $(F \vee G)$, $(F \dot{\vee} G)$, $(F \rightarrow G)$, $(F \leftrightarrow G)$ *irgi formulės.*

Formulių pavyzdžiai: $((p \dot{\vee} q) \rightarrow \neg(\neg p \vee q))$, $\neg p$, $((p_1 \leftrightarrow p_2) \vee \neg\neg(p_1 \& (p_3 \vee \neg p_1)))$.

Paprastumo dėlei, išorinių skliaustų formulėse nerašysime. Iš apibrėžimo išplaukia, kad kiekvienai formulei, jei ji nėra loginis kintamasis, arba atsiras tokia formulė G, kad F yra pavidalo $\neg G$ (tuo atveju rašysime $F = \neg G$ arba $F : \neg G$), arba atsiras dvi tokios formulės G, H, kad $F = G \alpha H$ ($\alpha \in \{\&, \vee, \dot{\vee}, \rightarrow, \leftrightarrow\}$). Pirmuoju atveju **pagrindine formulės F logine operacija** vadinsime $\neg$, o antruoju $- \alpha$.

Pavyzdžiai:
1. Formulės $(p \rightarrow q) \& (q \vee \neg(q \rightarrow p))$ pagrindinė loginė operacija yra &.

2. Formulės $\neg((p \to q) \vee (\neg q \to \neg p))$ pagrindinė loginė operacija yra $\neg$.

2.2 apibrėžimas.
1. *Formulė F yra F poformulis.*
2. *Jei $F = \neg G$, tai formulė G ir visi G poformuliai yra ir F poformuliai.*
3. *Jei $F = G\alpha H$ ($\alpha \in \{\&, \vee, \dot{\vee}, \to, \leftrightarrow\}$), tai formulės G, H bei jų poformuliai yra ir F poformuliai.*

Kai kada formules žymėsime pavidalu $F(p_1, ..., p_n)$. Tuo atveju $p_1, ..., p_n$ yra pilnas sąrašas skirtingų loginių kintamųjų, kurie yra formulės F poformuliai.

Pavyzdžiai:
1. Jei F yra $(p \to \neg p) \vee \neg\neg p$, tai $F = F(p)$,
2. Jei F yra $(p \vee q) \to (\neg r \vee p)$, tai $F = F(p, q, r)$.

Skirsime poformulio ir **poformulio įeities** sąvokas. Formulė yra žodis tam tikroje abėcėlėje. Todėl sąvoka *žodžio įeitis* perkeliama į *poformulio įeitis*.

Pavyzdys.
$F = (\mathbf{p\&\neg q}) \to (\neg(\mathbf{p\&\neg q}) \vee ((q \to (\mathbf{p\&\neg q}))\dot{\vee}(\neg p\&(\mathbf{p\&\neg q}))))$.
Formulėje F yra 4 poformulio $p\&\neg q$ įeitys (jos paryškintos) – pirmoji, antroji, trečioji ir ketvirtoji (iš kairės į dešinę).
Formulėje F yra 5 poformulio q įeitys.

Pavyzdys. Formulės $(p \to \neg q)\&\neg(q \vee (p \vee \neg r))$ poformuliais yra: $(p \to \neg q)\&\neg(q \vee (p \vee \neg r))$, $p \to \neg q$, $\neg(q \vee (p \vee \neg r))$, p, $\neg q$, q, $q \vee (p \vee \neg r)$, $(p \vee \neg r)$, $\neg r$, r.

Panašiai suprantamos ir loginių operacijų ar skliaustų įeitys.

2.2 Ekvivalenčios formulės

2.3 apibrėžimas. Interpretacija *loginių kintamųjų aibei $\{p_1, ..., p_n\}$ vadinsime kurią nors funkciją ν su apibrėžimo sritimi $\{p_1, ..., p_n\}$ ir rikšmėmis iš $\{t, k\}$.*

Turėdami interpretaciją ν aibei $\{p_1, ..., p_n\}$ ($\nu(p_i) = \beta_i$ ($\beta_i \in \{t, k\}$)) galim apskaičiuoti formulės $F(p_1, ..., p_n)$ reikšmę. Apskaičiuodami naudojames loginių operacijų apibrėžimo lentelėmis. Skliaustai nurodo operacijų atlikimo tvarką. Jei formulės reikšmė lygi β ($\beta \in \{t, k\}$), tai rašome $\nu(F) = \beta$ arba tiesiog $F = \beta$. Jei formulę sudaro n skirtingų loginių kintamųjų, tai yra 2^n skirtingų interpretacijų.

Sudarysime formulės $\neg p \to (p\&(q \vee \neg r))$ **teisingumo lentelę** (apskaičiuosime formulės reikšmes visoms galimoms interpretacijoms).

p,	q,	r	$\neg p$	$\neg r$	$q \vee \neg r$	$p\&(q \vee \neg r)$	$\neg p \to (p\&(q \vee \neg r))$
t	t	t	k	k	t	t	t
t	t	k	k	t	t	t	t
t	k	t	k	k	k	k	t
t	k	k	k	t	t	t	t
k	t	t	t	k	t	k	k
k	t	k	t	t	t	k	k
k	k	t	t	k	k	k	k
k	k	k	t	t	t	k	k

2.4 apibrėžimas. *Dvi formules $F(p_1, ..., p_n)$, $G(p_1, ..., p_n)$ vadinsime* **ekvivalenčiomis** *(rašysime $F(p_1, ..., p_n) \equiv G(p_1, ..., p_n)$), jei su bet kuria interpretacija ν $\nu(F) = \nu(G)$.*

Atkreipiame dėmesį, kad $\equiv$ nėra loginė operacija. Ekvivalenčių formulių pavyzdžiai:

$$p\&q \equiv q\&p$$
$$p\&(q\&r) \equiv (p\&q)\&r$$
$$p \vee q \equiv q \vee p$$
$$p \vee (q \vee r) \equiv (p \vee q) \vee r$$
$$\neg\neg p \equiv p$$
$$p \vee q \equiv \neg p \to q$$
$$p \dot{\vee} q \equiv (p \vee q)\&\neg(p\&q)$$

Naudodamiesi antraja bei ketvirtaja pora ekvivalenčių formulių, mes praleidžiame skliaustus, kai turime formules pavidalo $F_1\&...\&F_n$ arba $F_1 \vee ... \vee F_n$, nes tokių formulių reikšmės nepriklauso nuo suskliaudimo tvarkos. Ekvivalentumas aprašytųjų formulių tikrinamas sudarant jų teisingumo lenteles. Pavyzdžiui, parodysime, kad $p \vee q \equiv \neg p \to q$.

p,	q	$\neg p$	$p \vee q$	$\neg p \to q$
t	t	k	t	t
t	k	k	t	t
k	t	t	t	t
k	k	t	k	k

Kaip matome, reikšmių stulpeliai, atitinkantys $p \vee q$ ir $\neg p \to q$ sutampa. Kai kuriomis poromis ekvivalenčių formulių mes dažnai naudosimės. Išvardinsime jas bei priskirsime numerius.

$$p \to q \equiv \neg p \vee q \tag{2.1}$$

$$\neg(p\&q) \equiv \neg p \vee \neg q \tag{2.2}$$

$$\neg(p \vee q) \equiv \neg p\&\neg q \tag{2.3}$$

$$p \leftrightarrow q \equiv (p \to q)\&(q \to p) \tag{2.4}$$

$$(p\&q) \lor r \equiv (p \lor r)\&(q \lor r) \quad (2.5)$$

$$(p \lor q)\&r \equiv (p\&r) \lor (q\&r) \quad (2.6)$$

Ekvivalenčių formulių savybės:
1. Jei $A \equiv B$, tai ir $B \equiv A$,
2. Jei $A \equiv B$ ir $B \equiv C$, tai ir $A \equiv C$,
3. $A \equiv B$ tada ir tiktai tada, kai $\neg A \equiv \neg B$.

Apibendrinsime ekvivalentumo sąvoką, kai formulėse yra ne tik vienodi loginiai kintamieji.

2.5 apibrėžimas. *Dvi formulės $A(p_1, ..., p_n, q_1, ..., q_s)$ ir $B(p_1, ..., p_n, r_1, ..., r_u)$ vadinamos ekvivalenčiomis, jei su bet kuria interpretacija ν aibei $\{p_1, ..., p_n, q_1, ..., q_s, r_1, ..., r_u\}$, $\nu(A) = \nu(B)$.*

Pavyzdžiai.
1. $p \lor (q\&\neg q) \equiv \neg\neg p \lor (r\&\neg r)$.
2. $(p\&q) \to (\neg p \lor q) \equiv p \to p$.

Jei formulės $A(p_1, ..., p_n, q_1, ..., q_s)$ ir $B(p_1, ..., p_n)$ ekvivalenčios, tai loginiai kintamieji $q_1, ..., q_s$ vadinami *fiktyviais* formulėje A. Pateiksime dar vieną fiktyvaus loginio kintamojo apibrėžimą.

2.6 apibrėžimas. *Kintamasis p_i formulėje $A(p_1, ..., p_{i-1}, p_i, p_{i+1}, ..., p_n)$ vadinamas* **fiktyviu**, *jei su bet kuria interpretacija ν aibei $\{p_1, ..., p_{i-1}, p_{i+1}, ..., p_n\}$ $(\nu(p_j) = \beta_j)$ $\nu(A(\beta_1, ..., \beta_{i-1}, t, \beta_{i+1}, ..., \beta_n)) = \nu(A(\beta_1, ..., \beta_{i-1}, k, \beta_{i+1}, ..., \beta_n))$. Priešingu atveju, p_i vadinamas* **esminiu**.

Tarkime, A yra formulės F poformulis. Pažymėkime $F(B/A)$ formulę, gauta pakeitus formulėje F visas A įeitis formule B.

2.1 teorema. *Jei $A(p_1, ..., p_n)$ yra formulės F poformulis ir $A(p_1, ..., p_n) \equiv B(p_1, ..., p_n)$, tai $F \equiv F(B/A)$.*

Įrodymas. Sudarykime F teisingumo lentelę taip, kad iš pradžių joje būtų apskaičiuojamos A reikšmės (prieš tai, apskaičiuojant A poformulių reikšmes, t.y. tų formulių, kurios reikalingos A reikšmei nustatyti). Analogiškai sudarykime $F(B/A)$ teisingumo lentelę taip, kad iš pradžių būtų apskaičiuojamos B reikšmės. Tada gali skirtis F ir $F(B/A)$ teisingumo reikšmės tik iki stulpelių A, B, o visos reikšmės, esančios dešinėje tų stulpelių, sutampa. Tuo pačiu sutampa ir formulių F bei $F(B/A)$ reikšmės. Teorema įrodyta.

1 išvada. *Jei $A(p_1, ..., p_n, q_1, ..., q_s)$ yra formulės F poformulis ir $A(p_1, ..., p_n, q_1, ..., q_s) \equiv B(p_1, ..., p_n, r_1, ..., r_u)$, tai $F \equiv F(B/A)$.*

2 išvada. *Tarkime A yra F poformulis ir $A \equiv B$. Pakeitę kai kurias A įeitis formule B, gauname formulę ekvivalenčią F.*

3 išvada. *Jei $F(p_1, ..., p_n) \equiv G(p_1, ..., p_n)$ ir $A_1, ..., A_n$ bet kokios formulės, tai ir $F(A_1/p_1, ..., A_n/p_n) \equiv G(A_1/p_1, ..., A_n/p_n)$.*

Visos išvados įrodomos panašiai kaip ir 2.1 teorema, nagrinėjant atitinkamų formulių teisingumo lenteles.

2.6 apibrėžimas. *Loginių operacijų aibė E vadinama pilna, jei kiekvienai formulei galima rasti ekvivalenčią, kurioje yra tik loginės operacijos iš aibės E.*

2.2 teorema. *Aibės $\{\neg, \&\}$, $\{\neg, \vee\}$, $\{\neg, \rightarrow\}$ yra pilnos.*

Įrodymas.
1. Įrodysime, kad $\{\neg, \&\}$ pilna. Remiantis 2.1 teorema bei išvadomis, pakanka parodyti, kad formulėms $p \vee q$, $p \dot{\vee} q$, $p \rightarrow q$, $p \leftrightarrow q$ galima rasti ekvivalenčias, kuriose yra tik loginės operacijos iš $\{\neg, \&\}$. Jas rasime naudodamiesi tik aukščiau aprašytomis ekvivalenčių formulių poromis bei jų savybėmis.
$p \vee q \equiv \neg(\neg p \& \neg q)$
$p \dot{\vee} q \equiv (p \vee q) \& \neg(p \& q) \equiv \neg(\neg p \& \neg q) \& \neg(p \& q)$
$p \rightarrow q \equiv \neg p \vee q \equiv \neg(p \& \neg q)$
$p \leftrightarrow q \equiv (p \rightarrow q) \& (q \rightarrow p) \equiv \neg(p \& \neg q) \& \neg(q \& \neg p)$

2. Nagrinėjame $\{\neg, \vee\}$.
$p \& q \equiv \neg(\neg p \vee \neg q)$
$p \dot{\vee} q \equiv (p \vee q) \& \neg(p \& q) \equiv \neg(\neg(p \vee q) \vee \neg(\neg p \vee \neg q))$
$p \rightarrow q \equiv \neg p \vee q$
$p \leftrightarrow q \equiv (p \rightarrow q) \& (q \rightarrow p) \equiv (\neg p \vee q) \& (\neg q \vee p) \equiv \neg(\neg(\neg p \vee q) \vee \neg(\neg q \vee p))$

3. Nagrinėjame $\{\neg, \rightarrow\}$.
$p \vee q \equiv \neg p \rightarrow q$
$p \& q \equiv \neg(\neg p \vee \neg q) \equiv \neg(p \rightarrow \neg q)$
$p \dot{\vee} q \equiv (p \vee q) \& \neg(p \& q) \equiv (\neg p \rightarrow q) \& (p \rightarrow \neg q) \equiv \neg((\neg p \rightarrow q) \rightarrow \neg(p \rightarrow \neg q))$
$p \leftrightarrow q \equiv (p \rightarrow q) \& (q \rightarrow p) \equiv \neg((p \rightarrow q) \rightarrow \neg(q \rightarrow p))$
Teorema įrodyta.

Yra ir kiti pilni loginių operacijų poaibiai, pavyzdžiui, $\{\dot{\vee}, \rightarrow\}$. Bet visuose juose yra ne mažiau kaip du elementai. Galima rasti ir vieną tokią loginę operaciją, kuri sudarytų pilną aibę. Pirmasis tokią operaciją sugalvojo 1880 metais Ch. Peirce, bet jis neskyrė tam ypatingo dėmesio. Jo darbas nebuvo publikuotas, o pati operacija užmiršta. Žymiai vėliau, 1913 metais kitą tokią operaciją aprašė H.M.Sheffer (mes ją vadinsim Sheffer funkcija). Tiesa, visoms mūsų nagrinėjamoms loginėms operacijoms yra loginių jungčių, vartojamų šnekamojoje kalboje, atitikmenys *ir, arba,...* (iš čia jos ir kilusios). Atitikmenų Peirce bei Sheffer opercijoms šnekamojoje kalboje nėra.

Peirce operaciją žymėsime $\uparrow$, o Sheffer funkciją $-$ $|$. Jos nusakomos tokiomis lentelėmis:

p,	q	$p \uparrow q$
t	t	k
t	k	k
k	t	k
k	k	t

p,	q	$p\|q$
t	t	k
t	k	t
k	t	t
k	k	t

Jos sudaro pilnąsias aibes (išplaukia iš 2.2 teoremos), nes $\neg p \equiv p \uparrow q$ ir $p \vee q \equiv (p \uparrow q) \uparrow (p \uparrow q)$, bei $\neg p \equiv p|p$ ir $p\&q \equiv (p|q)|(p|q)$.

Logikoje yra dvi konstantos $-$ **tiesa** (t) ir **melas** (k). Formulėse jos nesutinkamos išreikštiniu pavidalu, nes jas galima eliminuoti, t.y. rasti kitą, ekvivalenčią formulę be loginių konstantų įeičių. Remiames 2.1 teorema su išvadomis bei loginių

2.7 apibrėžimas. *Formulė F vadinama* **tapačiai teisinga**, *jei su bet kuria interpretacija ν $\quad \nu(F) = t$.*

Tapačiai teisingos formulės dar vadinamos **tautologijomis** arba **logikos dėsniais.** Jos vaidina ypatingai svarbų vaidmenį logikoje, nes daugumos uždavinių sprendimą pavyksta suvesti į kurios nors formulės tapataus teisingumo nustatymą. Tapačiai teisingų formulių pavyzdžiai:

$$\begin{gathered}
\neg\neg p \to p \\
(p \& q) \to p \\
(p \to q) \to (\neg q \to \neg p) \\
p \to (p \vee q)
\end{gathered}$$

Norint patikrinti ar formulė $A(p_1, ..., p_n)$ yra tapačiai teisinga, pakanka sudaryti jos teisingumo lentelę ir pažiūrėti, ar paskutiniame stulpelyje vien tik reikšmės t. Bet sudarant teisingumo lentelę reikia apskaičiuoti formulės A reikšmę 2^n interpretacijoms. Tai labai greitai auganti funkcija, todėl praktiškai galime naudotis teisingumo lentele, nustatant formulės tapačiai teisingumą tik nedideliems n.

2.3 teorema. *Tarkime $F(p_1, ..., p_n)$ yra tapačiai teisinga formulė, o $A_1, ..., A_n$ – bet kurios formulės. Tuomet $F(A_1, ..., A_n)$ yra tapačiai teisinga formulė.*

Įrodymas. Priskirkime visiems loginiams kintamiesiems, įeinantiems į $A_1, ..., A_n$ bet kurias reikšmes. Tarkime, esant toms reikšmėms $A_1 = \alpha_1, ..., A_n = \alpha_n$ (čia $\alpha_i \in \{t, k\}$; $(1 \leq i \leq n)$). Tada formulės $F(A_1, ..., A_n)$ reikšmė, esant toms pačioms loginių kintamųjų reikšmėms, sutaps su $F(\alpha_1, ..., \alpha_n)$. Kadangi $F(p_1, ..., p_n)$ reikšmė, su bet kokiomis loginių kintamųjų $p_1, ..., p_n$ reikšmėmis, lygi t, tai ji bus lygi t, ir kai $p_1 = \alpha_1, ..., p_n = \alpha_n$. Teorema įrodyta.

2.8 apibrėžimas. *Formulė F vadinama* **tapačiai klaidinga**, *jei su bet kuria interpretacija ν $\quad \nu(F) = k$.*

Tapačiai klaidingų formulių pavyzdžiai:

$$\begin{gathered}
p \& \neg p \\
\neg((p \& q) \to p) \\
(p \to q) \& \neg((q \to r) \to (p \to r))
\end{gathered}$$

Tapačiai klaidingų formulių savybės:
1. F tapačiai klaidinga tada ir tiktai tada, kai $\neg F$ tapačiai teisinga formulė,
2. Jei F tapačiai klaidinga formulė, o G – bet kuri formulė, tai $F \to G$ yra tapačiai teisinga formulė.
3. Tapačiai klaidingų bei tapačiai teisingų formulių visi loginiai kintamieji yra fiktyvūs.

2.9 apibrėžimas. *Formulė F vadinama* **įvykdoma**, *jei atsiras tokia interpretacija ν, su kuria $\nu(F) = t$.*

Iš apibrėžimo išplaukia, kad kiekviena tapačiai teisinga formulė yra tuo pačiu ir įvykdoma formulė. Suprantama, yra įvykdomų formulių, kurios nėra tapačiai teisingos. Pavyzdžiui, $p \to q$, $p \to (p\&q)$, $p \leftrightarrow \neg q$.

2.10 apibrėžimas. *Sakysime, kad formulė F yra formulių aibės $A = \{F_1, ..., F_n\}$* **loginė išvada**, *jei su bet kuria interpretacija ν, su kuria visos aibės A formulės teisingos, teisinga yra ir F (t.y. $\nu(F) = t$).*

Dažniausiai aibės A formulės vadinamos *prielaidomis* arba *tai, kas duota.* Formulė F vadinama *išvada, tikslu* arba *tai, ką reikia įrodyti.* Kai kada, priklausomai nuo užduoties, A vadinama *žinių baze*, o F – *užklausa.* Taip pat, vietoje loginė išvada, vartojamas terminas *samprotavimas teisingas* arba *samprotavimas pagrįstas.* Žymima ir taip

$$\frac{\begin{array}{c} F_1 \\ \vdots \\ F_n \end{array}}{F}$$

Iš apibrėžimo išplaukia, kad F yra $\{F_1, ..., F_n\}$ loginė išvada tada ir tiktai tada, kai $(F_1\&....\&F_n) \to F$ yra tapačiai teisinga formulė.

Pavyzdžiai.

1. Jei vaikas netvarkingas, tai jis blogai mokosi. Vadinasi, jei vaikas blogai mokosi, tai jis netvarkingas.

Formalizacija. Pažymėkime raide n teiginį "vaikas netvarkingas", o raide b – "vaikas blogai mokosi". Tuomet turime

$$\frac{n \to b}{b \to n}$$

Formulė $(n \to b) \to (b \to n)$ nėra tapačiai teisinga, nes su interpretacija $n = k$, $b = t$, ji klaidinga. Taigi, iš prielaidos $n \to b$ neišplaukia $b \to n$.

2. Kiekvienas aibės A elementas nepriklauso aibei D. Jei x (bet kuris) nepriklauso aibei B, tai jis nepriklauso ir aibei C. Jei x priklauso aibei B, tai jis priklauso ir A, C sankirtai. Vadinasi, jei e yra aibės C elementas, tai jis nėra D elementas.

Formalizacija. Pažymėkime raide a teiginį $e \in A$, raide b – $e \in B$, c – $e \in C$ ir d – $e \in D$. Tuomet turime

$$\begin{array}{c} a \to \neg d \\ \neg b \to \neg c \\ b \to (a\&c) \\ \hline c \to \neg d \end{array}$$

Formulė $((a \to \neg d)\ \&\ (\neg b \to \neg c)\ \&\ (b \to (a\&c))) \to (c \to \neg d)$ yra tapačiai teisinga. Tuo galime įsitikinti sudarę teisingumo lentelę. Vadinasi, formulė $c \to \neg d$ yra duotųjų (virš brūkšnio) išvada, o samprotavimas, aprašytas užduotyje, teisingas.

2.4 Normaliosios formos

Dažniausiai nagrinėjamos ne bet kurio, o tam tikro pavidalo formulės, vadinamos *normaliosios formos.* Skirsime dvi normaliąsias formas: *normaliąją disjunkcinę formą* (sutrumpintai žymėsime NDF) ir *normaliąją konjunkcinę formą* (žymėsime NKF). Parodysime, kad kiekvienai formulei galima rasti ekvivalenčią kaip normaliosios disjunkcinės, taip ir normaliosios konjunkcinės formos.

2.11 apibrėžimas. Litera *vadinamas loginis kintamasis bei loginio kintamojo neigimas.*

2.12 apibrėžimas. Disjunktu *vadiname literų disjunkciją* $l_1 \vee l_2 \vee ... \vee l_v$, *o skaičių* v *jo ilgiu.*

2.13 apibrėžimas. Konjunktu *vadiname literų konjunkciją* $l_1\&l_2\&...\&l_v$, *o skaičių* v *jo ilgiu.*

Disjunkto (konjunkto) D ilgį žymėsime $i(D)$.

Pavyzdžiai.
Šios formulės yra disjunktai:
$p \vee \neg q \vee \neg r$, $\neg p_1 \vee p_2 \vee \neg p_3 \vee p_4$, p, $\neg p$, $\neg p \vee q$.

Šios formulės yra konjunktai:
$p\&\neg q$, p, $\neg q$, $\neg p\&q\&\neg p\&\neg r$.

Laikysime, kad kaip disjunktuose, taip ir konjunktuose yra tik po vieną skirtingų literų įeitį. Naudosimes tuo, kad $l \vee l \equiv l$, $l\&l \equiv l$ (l – kuri nors litera). Pavyzdžiui, vietoje disjunkto $p \vee p \vee \neg r \vee \neg q \vee \neg q \vee \neg r$ nagrinėsime jam ekvivalentų $p \vee \neg q \vee \neg r$.

2.14 apibrėžimas. *Formulės* **normaliaja disjunkcine forma** *vadinsime jai ekvivalenčią formulę pavidalo* $\vee_{i=1}^{s} K_i$. *Čia* K_i $(i = 1, ..., s)$ – *konjunktai.*

NDF pavyzdžiai:
$(p\&q) \vee (p\&\neg q\&r) \vee \neg p \vee (q\&\neg p)$,

$p\&q\&\neg r$ (šiuo atveju $s=1$),
$p \vee \neg q \vee \neg r$.

2.4 teorema. *Kokia bebūtų formulė, galima rasti jai ekvivalenčią normaliosios disjunkcinės formos.*

Įrodymas. Aprašysime du transformavimo į NDF būdus.

a) *Transformavimas į NDF naudojantis teisingumo lentele.*

Tarkime duota formulė $F(p_1,...,p_n)$. Sudarome jos teisingumo lentelę. Tarkime, F nėra tapačiai klaidinga. Kiekvienai interpretacijai ν, su kuria F teisinga, priskiriame po konjunktą $l_1\&l_2\&...\&l_n$.

$$l_i = \begin{cases} p_i, & jei \quad \nu(p_i)=t \\ \neg p_i & jei \quad \nu(p_i)=k \end{cases}$$

Nesunku matyti, kad konjunktas, atitinkantis ν, teisingas tik su vienintele interpretacija ν. Disjunkcija tokių konjuktų ir bus NDF ekvivalenti formulei $F(p_1,...,p_n)$. Jei F tapačiai klaidinga, tai jos NDF laikysime $p_1\&\neg p_1$.

Pavyzdys.

p, q, r	F(p,q,r)
t t t	k
t t k	t
t k t	t
t k k	k
k t t	k
k t k	t
k k t	k
k k k	k

Ji teisinga su trimis interpretacijomis:
1) $p=q=t$, $r=k$,
2) $p=r=t$, $q=k$,
3) $p=r=k$, $q=t$.
Pirmajai interpretacijai priskiriame $p\&q\&\neg r$, antrajai – $p\&\neg q\&r$, trečiajai – $\neg p\&q\&\neg r$. Formulės NDF bus $(p\&q\&\neg r) \vee (p\&\neg q\&r) \vee (\neg p\&q\&\neg r)$.

b) *Transformavimas į NDF naudojantis ekvivalenčiomis formulėmis.*

Transformavimo algoritmas susideda iš tokių žingsnių:

1) *Eliminavimas.* Keičiame visus poformulius $G \to H$ pavidalo (primename, kad nagrinėjame formules, kuriose sutinkamos tik loginės operacijos $\neg, \&, \vee, \to$) jiems ekvivalenčiais, kuriuose yra tik $\neg, \&, \vee$. Naudojames (2.1) ekvivalentumu.

2) *Neigimo įkėlimas į skliaustus.* Naudojantis (2.2), (2.3) ekvivalentumais bei $\neg\neg G \equiv G$, galime pasiekti, kad neigimas formulėje būtų tik prieš loginius kintamuosius.

3) *Distributyvumo dėsnio taikymas.* Taikant (2.6) ekvivalentumą, gauname formulę, kuri ir bus NDF.

4) *Prastinimas.* Taikome ekvivalentumus $G \vee G \equiv G$, $G\&G \equiv G$, $\neg\neg G \equiv G$. Poformulius $G\&\neg G$, $G \vee \neg G$ keičiame atitinkamai konstantomis k, t bei jas eliminuojame. Prastinimą galima taikyti ir po pirmojo bei antrojo žingsnių.
Teorema įrodyta.

Pavyzdys. Transformuosime $\neg(p \to q) \vee (q\&r)$ į NDF.
Eliminuojame $\to$. Gauname $\neg(\neg p \vee q) \vee (q\&r)$.
Įkeliame neigimą į skliaustus. $(p\&\neg q) \vee (q\&r)$.
Taikome distributyvumo dėsnį. $(p\&q) \vee (p\&r) \vee (\neg q\&q) \vee (\neg q\&r)$.
Prastiname ir gauname NDF $(p\&q) \vee (p\&r) \vee (\neg q\&r)$.

2.15 apibrėžimas. *Formulės* **normaliaja konjunkcine forma** *vadinsime jai ekvivalenčią formulę pavidalo* $\&_{i=1}^{s} D_i$. *Čia* D_i $(i = 1, ..., s)$ $-$ *disjunktai.*

NKF pavyzdžiai:
$(p \vee \neg q)\&(p \vee \neg q \vee r)\&\neg p\&(p \vee q)$,
$p \vee \neg q \vee \neg r$ (šiuo atveju $s = 1$),
$p\&q\&\neg r$.

Kaip matome, pastarosios dvi formulės yra kaip normaliosios konjunkcinės, taip ir normaliosios disjunkcinės formos.

2.5 teorema. *Kokia bebūtų formulė, galima rasti jai ekvivalenčią normaliosios konjunkcinės formos.*

Įrodymas. Kaip ir 2.4 teoremos įrodyme, aprašysime du transformavimo į NKF būdus.

a) *Transformavimas į NKF naudojantis teisingumo lentelėmis.*

Tarkime duota formulė $F(p_1, ..., p_n)$. Sudarome jos teisingumo lentelę. Be to, tarkime F nėra tapačiai teisinga. Kiekvienai interpretacijai ν, su kuria F klaidinga, priskiriame po disjunktą $l_1 \vee l_2 \vee ... \vee l_s$.

$$l_i = \begin{cases} \neg p_i, & jei \quad \nu(p_i) = t \\ p_i, & jei \quad \nu(p_i) = k \end{cases}$$

Nesunku matyti, kad disjunktas, atitinkantis ν, klaidingas tik su vienintele interpretacija ν. Konjunkcija tokių disjunktų ir bus NKF ekvivalenti formulei $F(p_1, ..., p_n)$. Jei F tapačiai teisinga, tai jos NKF laikysime $p_1 \vee \neg p_1$.

Pavyzdys.

p, q, r	F(p,q,r)
t, t, t	t
t, t, k	t
t, k, t	k
t, k, k	t
k, t, t	t
k, t, k	k
k, k, t	k
k, k, k	t

Formulė klaidinga su trimis interpretacijomis:
1) $p = r = t$, $q = k$,
2) $p = r = k$, $q = t$,
3) $p = q = k$, $r = t$.
Pirmajai interpretacijai priskiriame disjunktą $\neg p \vee q \vee \neg r$, antrajai – $p \vee \neg q \vee r$, trečiajai – $p \vee q \vee \neg r$. Formulės NKF bus $(\neg p \vee q \vee \neg r) \& (p \vee \neg q \vee r) \& (p \vee q \vee \neg r)$.

b) *Transformavimas į NKF naudojantis ekvivalenčiomis formulėmis.*

Transformavimo algoritmo *eliminavimo, neigimo įkėlimo į skliaustus* bei *prastinimo* žingsniai yra tokie patys, kaip kad aprašyti 2.4 teoremos įrodyme, transformuojant formulę į NDF naudojantis ekvivalenčiomis formulėmis. Skiriasi nuo aprašytojo algoritmo tik trečiuoju žingsniu t.y. distributyvumo dėsnio taikymu. Šiuo atveju naudojames (2.6) ekvivalentumu. Teorema įrodyta.

Iš loginių kintamųjų $p_1, ..., p_n$ galima sudaryti 3^n skirtingų disjunktų (įskaitant ir atitinkantį tapačiai klaidingą formulę). Todėl skirtingų NDF bus 2^{3^n}. Formulės turi ne vieninteles normaliąsias disjunkcines bei konjunkcines formas. Pavyzdžiui, formulės $F(p, q, r)$, aprašytosios, po 2.4 teoremos įrodymo, pavyzdyje, NDF yra $(p \& q \& \neg r) \vee (p \& \neg q \& r) \vee (\neg p \& q \& \neg r)$, o taip pat ir $(q \& \neg r) \vee (p \& \neg q \& r)$. Todėl NDF bei NKF dar skirstomos į *tobulas, trumpiausias* bei *minimalias.*

2.16 apibrėžimas. *Formulės* $F(p_1, ..., p_n)$ **normalioji disjunkcinė forma** $\vee_{i=1}^{s}$ $(i = 1, ..., s)$ *vadinama* **tobula**, *jei kiekviename konjunkte* K_i $(i = 1, ..., s)$ *sutinkamas arba* p_j, *arba* $\neg p_j$ $(j = 1, ..., n)$.

2.17 apibrėžimas. *Formulės* $F(p_1, ..., p_n)$ **NDF** $\vee_{i=1}^{s} K_i$ *vadinama* **trumpiausia**, *jei bet kuri kita jos NDF* $\vee_{i=1}^{m} K_i'$ *tenkina sąlygą:* $m \geq s$.

2.18 apibrėžimas. *Formulės* $F(p_1, ..., p_n)$ **NDF** $\vee_{i=1}^{s} K_i$ *vadinama* **minimalia**, *jei bet kuri kita jos NDF* $\vee_{i=1}^{m} K_i'$ *tenkina sąlygą:* $i(K_1) + ... + i(K_s) \leq i(K_1') + ... + i(K_m')$.

Panašiai apibrėžiamos ir *tobulos, trumpiausios* bei *minimalios* NKF.

2.5 Logikos algebros funkcijos

Aprašysime dar vieną formulių pavidalą, į kurį galima transformuoti bet kurią formulę. Šiame skyrelyje vietoje konstantos t naudosime 1, o vietoje k – 0, bei, kaip įprasta logikos algebros funkcijoms, griežtąją disjunkciją (sudėtį moduliu 2) žymėsime $\oplus$, o konjunkciją – $\cdot$ (sandauga).

p, q	$p \oplus q$
1 1	0
1 0	1
0 1	1
0 0	0

p, q	$p \cdot q$
1 1	1
1 0	0
0 1	0
0 0	0

2.19 apibrėžimas. *Funkcijos, kurių apibrėžimo ir kitimo sritys yra aibė* $\{0, 1\}$, *vadinamos* **logikos algebros funkcijomis** *(arba Boole funkcijomis).*

Formulės yra taip pat ir logikos algebros funkcijos. Logikos algebros funkcijas galima nusakyti lentelėmis:

$p_1, \cdots, p_{n-1}, p_n$	$f(p_1, ..., p_{n-1}, p_n)$
$0 \cdots 0\ 0$	$f(0, ..., 0, 0)$
$0 \cdots 0\ 1$	$f(0, ..., 0, 1)$
$0 \cdots 1\ 0$	$f(0, ..., 1, 0)$
$\cdots$	$\cdots$
$1 \cdots 1\ 1$	$f(1, ..., 1, 1)$

Dvi logikos algebros funkcijos vadinamos lygiomis, jeigu jas atitinkančios lentelės yra lygios. Kiekvieną logikos algebros funkciją galima išreikšti formule t.y. rasti tokią formulę, kad jų lentelės būtų vienodos. Tai išplaukia iš praeito skyrelio. Juk kiekvieną logikos algebros funkciją galima transformuoti į normaliąją disjunkcinę (konjunkcinę) formą. Taigi, logikos algebros funkcijos užrašymui formule pakanka loginių operacijų $\neg$, &, $\vee$. Paprastumo dėlei vietoj *logikos algebros funkcija* šiame skyrelyje rašysime *funkcija.*

Aibė $\{0, 1, p \cdot q, p \oplus q\}$ yra pilna t.y. kokia bebūtų funkcija (o tuo pačiu ir formulė) galima rasti jai lygią, kurioje yra tik sandauga, sudėtis (moduliu 2) bei konstanta

1. Tai išplaukia iš to, kad $\{\neg, \&\}$ yra pilna aibė ir $p\&q = p \cdot q$ (rašysime paprasčiau pq), o $\neg p = p \oplus 1$. Formulėje $F(p_1, ..., p_s)$, į kurią įeina tik konstanta 1, sandauga bei sudėtis moduliu 2, atlikę algebrinius pertvarkius, gauname polinomą t.y. formulę, atrodančią šitaip ($p^1 = p$, o p^0 reiškia, kad naryje nėra p):

$$\sum_{i_1 \dots i_s} a_{i_1 \dots i_s} p_1^{i_1} \cdot p_2^{i_2} \cdot \dots \cdot p_s^{i_s} \qquad (a_{i_1 \dots i_s} \in \{0, 1\}) \tag{2.7}$$

Tokios formulės vadinamos **Žegalkino polinomais.** Keletas savybių:
$p^2 = pp = p\&p = p$,
$p^n = p$,
$p \oplus p = 0$,
$1 \oplus 1 = 0$.

Aprašysime du būdus, kaip bet kurią formulę $F(p_1, ..., p_n)$ transformuoti į Žegalkino polinomą.

1. Randame formulei $F(p_1, ..., p_n)$ ekvivalenčią, kurioje iš loginių operacijų tėra $\neg, \&$ (tokia aibė yra pilna). Konjunkciją keičiame sandauga, o neigimą – $\oplus 1$ (t.y. $\neg G = 1 \oplus G$). Atliekame algebrinius pertvarkius ir gauname Žegalkino polinomą.

Pavyzdys. Transformuoti $\neg(p\&q) \to r$ į Žegalkino polinomą.
Sprendimas. $\neg(p\&q) \to r \equiv (p\&q) \vee r \equiv \neg(\neg(p\&q)\&\neg r) \equiv 1 \oplus (1 \oplus pq)(1 \oplus r) \equiv 1 \oplus 1 \oplus pq \oplus r \oplus pqr$.
Taigi, formulės $\neg(p\&q) \to r$ Žegalkino polinomu yra $pqr \oplus pq \oplus r$.

2. *Neišreikštinių koeficientų metodas.*
Jei formulėje yra s loginių kintamųjų, tai užrašome bendrą (2.7) Žegalkino polinomo su s kintamaisiais pavidalą. Pagal formulės teisingumo lentelę sudarome 2^s lygčių sistemą, kurią išsprendę ir randame polinomo koeficientus.

Pavyzdys. Rasti formulei $p \to (q\&r)$ atitinkantį Žegalkino polinomą neišreikštinių koeficientų metodu.
Sprendimas. Formulėje yra trys loginiai kintamieji, todėl bendras Žegalkino polinomo su 3 kintamaisiais pavidalas bus:

$$a_1 pqr \oplus a_2 pq \oplus a_3 qr \oplus a_4 pr \oplus a_5 p \oplus a_6 q \oplus a_7 r \oplus a_8.$$

Sudarome teisingumo lentelę:

p, q, r	$q\&r$	$p \to (q\&r)$
1 1 1	1	1
1 1 0	0	0
1 0 1	0	0
1 0 0	0	0
0 1 1	1	1
0 1 0	0	1
0 0 1	0	1
0 0 0	0	1

$2^3 = 8$ lygčių sistema bus tokia:

$$\begin{cases} a_1 \oplus a_2 \oplus a_3 \oplus a_4 \oplus a_5 \oplus a_6 \oplus a_7 \oplus a_8 = 1 \\ \hfill a_2 \oplus a_5 \oplus a_6 \oplus a_8 = 0 \\ \hfill a_4 \oplus a_5 \oplus a_7 \oplus a_8 = 0 \\ \hfill a_5 \oplus a_8 = 0 \\ \hfill a_3 \oplus a_6 \oplus a_7 \oplus a_8 = 1 \\ \hfill a_6 \oplus a_8 = 1 \\ \hfill a_7 \oplus a_8 = 1 \\ \hfill a_8 = 1 \end{cases}$$

Išsprendę ją, gauname $a_8 = 1$, $a_7 = 0$, $a_6 = 0$, $a_3 = 0$, $a_5 = 1$, $a_4 = 0$, $a_2 = 0$, $a_1 = 1$. Taigi, formulę $p \to (q\&r)$ atitinkantis Žegalkino polinomas yra $pqr \oplus p \oplus 1$.

Atkreipiame dėmesį, kad kiekvienai formulei egzistuoja daug jos normaliųjų disjunkcinių (konjunkcinių) formų. Tuo tarpu kiekvienai formulei egzistuoja jai lygus tik vienas vienintelis Žegalkino polinomas. Tai išplaukia iš atitinkamos lygčių sistemos sprendinio vienatinumo.

2.6 Kai kurios neklasikinės logikos

Trumpai apžvelgsime dvi neklasikines logikas, o dar su dvejomis (intuicionistine bei modalumo logikomis) susipažinsime plačiau kituose skyriuose.

1. *Daugiareikšmė logika.* Nagrinėsime loginius kintamuosius, kurių kitimo sritimi yra natūraliųjų skaičių aibė $\{0, 1, ..., k-1\}$. Apibendrinsime kai kurias nagrinėtąsias logines operacijas, t.y. su $k = 2$ jų teisingumo lentelės sutaps su anksčiau aprašytomis.

a) *Neigimas.* Pateiksime tris skirtingus neigimo apibrėžimus k-reikšmės logikos atveju.

p	$\sim$ p
0	1
1	2
2	3
$\vdots$	$\vdots$
k-2	k-1
k-1	0

p	Np
0	k-1
1	k-2
2	k-3
$\vdots$	$\vdots$
k-2	1
k-1	0

$$I_i(p) = \begin{cases} k-1, & jei \quad p = i \\ 0, & jei \quad p \neq i \end{cases} \qquad (i = 0, ..., k-1)$$

b) *Konjunkcija.* $min(p, q)$ bei antras apibrėžimas $pq(mod \;\; k)$.

c) *Disjunkcija.* $max(p, q)$.

Nagrinėsime funkcijas, kurių apibrėžimo ir reikšmių kitimo sritimi yra $\{0, 1, ..., k-1\}$. Jas vadinsime k-reikšmėmis funkcijomis.
Loginių operacijų aibė vadinama *pilna*, jei kiekvieną k-reikšmę funkciją galima užrašyti formule, kurioje sutinkamos tik operacijos iš nagrinėjamosios aibės.

Aibė $\{0, 1, ..., k-1, I_0(p), ..., I_{k-1}(p), min(p, q), max(p, q)\}$ yra pilna. Pažymėkime $min(p, q)$ konjunkcijos simboliu, o $max(p, q)$ – disjunkcijos. Tuomet bet kuri k-reikšmė funkcija užrašoma formule:

$$f(p_1, ..., p_n) = \vee_{(\alpha_1 ... . \alpha_n)} I_{\alpha_1}(p_1) \& ... \& I_{\alpha_n}(p_n) \& f(\alpha_1, ..., \alpha_n) \quad \alpha_i \in \{0, 1, ..., k-1\}$$

2. *Netiksli logika.* Loginiai kintamieji įgyja kurią nors reikšmę (racionalųjį skaičių) iš intervalo [0,1]. Neigimas, konjunkcija bei disjunkcija apibrėžiami tokiu būdu:

$\neg p = 1 - p$,
$p \& q = min(p, q)$,
$p \vee q = max(p, q)$.

Iš apibrėžimo išplaukia, kad ne su visomis p reikšmėmis, $p \vee \neg p$ lygi 1, o $p \& \neg p$ lygi 0 t.y. pirmoji nėra tapačiai teisinga, o antroji tapačiai klaidinga įprastine prasme.

2.20 apibrėžimas. *Formulė $F(p_1, ..., p_n)$ tapačiai teisinga, jei su bet kuriomis reikšmėmis* $\alpha_1, ..., \alpha_n \in [0, 1]$ $F(\alpha_1, ..., \alpha_n) \geq 0,5$.

2.21 apibrėžimas. *Formulė $F(p_1, ..., p_n)$ tapačiai klaidinga, jei su bet kuriomis reikšmėmis* $\alpha_1, ..., \alpha_n \in [0, 1]$ $F(\alpha_1, ..., \alpha_n) \leq 0,5$.

$p \vee \neg p$ yra tapačiai teisinga formulė, nes $max(p, q) \geq 0,5$.
$p \& \neg p$ yra tapačiai klaidinga formulė, nes $min(p, \neg p) \leq 0,5$.

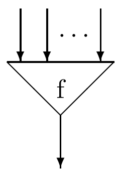
Implikacija $p \to q$ apibrėžiama tokiu būdu: $max(1 - p, q)$.

Įrodyta, kad *tapačiai teisingų formulių aibė sutampa su klasikinės logikos tapačiai teisingų formulių aibe.*

Formulės F, G vadinamos ekvivalenčiomis, jei $(F \to G)\&(G \to F)$ yra tapačiai teisinga formulė.
Formulės $\neg(p\&q)$ ir $\neg p \vee \neg q$ ekvivalenčios, nes $1 - min(p,q) = max(1-p, 1-q)$.
Formulės gali būti ekvivalenčios, bet su kai kuriomis kintamųjų reikšmėmis, jos gali nesutapti. Pavyzdžiui, $p\&q$ bei $(p\&q\&r) \vee (p\&q\&\neg r)$ yra ekvivalenčios, bet su $p = q = 0,9$, $r = 0,3$ jų reikšmės skiriasi.
$min(p,q) = 0,9$ t.y. $p\&q$ reikšmė lygi 0,9, o $min(p,q,r) = 0,3$, $min(p,q,\neg r) = 0,7$, t.y. $(p\&q\&r) \vee (p\&q\&\neg r)$ reikšmė šiuo atveju lygi 0,7.

2.7 Dvejetainis sumatorius

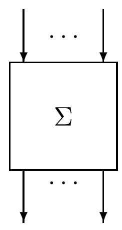
Nagrinėsime signalus transformuojančius elementus, turinčius n įėjimų $(n \geq 1)$ ir vieną išėjimą. Žymėsime juos schema



Signalai gali būti dviejų rūšių (žymėsime juos 0, 1). Laikome, kad elementai dirba diskrečiu režimu, t.y. signalai įėjime paduodami laikas nuo laiko vienu metu į visus įėjimo kanalus. Akimirksniu jie apdorojami elementu t.y. elemento darbo laikas artimas nuliui, ir išėjime gauname rezultatą (0 arba 1). Jei elementas, be to, tenkina sąlygą, kad jo darbas nepriklauso nuo praeities (elementas yra be atminties), tai jis vadinamas *loginiu*. Loginių elementų darbą galima aprašyti lentele

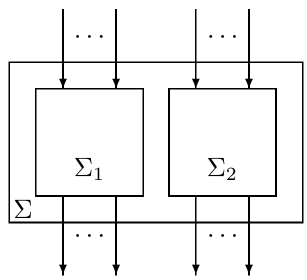
p_1, p_2,..., p_n	$f(p_1, p_2, ..., p_n)$
1 1 ... 1	f(1,1,...,1)
0 1 ... 1	f(0,1,...,1)
...	...
0 0 ... 0	f(0,0,...,0)

$p_1, ..., p_n$ žymi informaciją gaunamą n kanalais (iš kairės į dešinę), o $f(p_1, ..., p_n)$ – elemento darbo rezultatą. Sakysime, kad loginis elementas realizuoja logikos algebros funkciją $f(p_1, ..., p_n)$. Naudodami loginius elementus konstruosime schemas. Schemoje yra m įėjimų $(m \geq 1)$ ir n išėjimų $(n \geq 1)$.



Loginis elementas yra schema. Galimi trys schemų jungimo būdai.

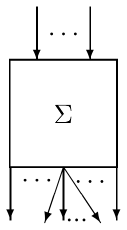
1. *Sujungimas.* Tarkime turime dvi schemas Σ_1, Σ_2 su m_1 ir m_2 įėjimais, bei n_1 ir n_2 išėjimais. Naujoje schemoje, gautoje sujungus duotąsias, bus $m_1 + m_2$ įėjimai ir $n_1 + n_2$ išėjimai.



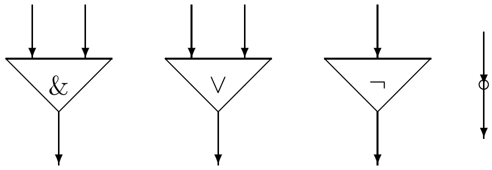
Schemos Σ_1, Σ_2 dirba lygiagrečiai ir tenkina sąlygą, kad informacija (signalai) įėjime į abi siunčiama vienu metu, t.y. darbas sinchronizuotas.

2. *Elemento prijungimas.* Tarkime turime schemą su m įėjimais, n išėjimais ir kurį nors elementą su k įėjimais ($k \leq n$) ir 1 išėjimu. Tuomet galima gauti naują schemą, prijungus elementą prie duotosios schemos išėjimų. Naujojoje schemoje bus m įėjimų ir $n - k + 1$ išėjimas.

3. *Išėjimų skaidymas.* Duota schema su m įėjimais ir n išėjimais. Galima gauti naują schemą išskaidžius kurį nors vieną išėjimą į k išėjimų.



Nagrinėsime schemas, kuriose yra tik keturių tipų loginiai elementai:



Paskutinysis elementas vadinamas trivialiu. Norime rasti schemą su $2n$ įėjimais ir $n+1$ išėjimu, kuri realizuotų dvejetainį sumatorių. T.y. schemos įėjime pateikiami du dvejetainiai skaičiai $x_n x_{n-1}...x_1$, $y_n y_{n-1}...y_1$ $(x_i, y_i \in \{0,1\})$, o išėjime, apdorojus informaciją, gaunama jų suma $z_{n+1}z_n...z_1$.

$$
\begin{array}{r}
+\begin{array}{r} x_n x_{n-1}...x_1 \\ y_n y_{n-1}...y_1 \end{array} \\
\hline
z_{n+1} z_n z_{n-1}...z_1
\end{array}
$$

Įvedame dar schemos vidinius kintamuosius $q_i \in \{0,1\}$, kurių prasmė tokia: $q_1 = 0$ ir $q_{i+1} = 1$ tada ir tiktai tada, kai $x_i + y_i + q_i > 1$. Sudedame tokius skaičius:

$$
\begin{array}{lr}
 & q_{n+1}q_n...q_1 \\
\oplus & x_n...x_1 \\
 & y_n...y_1 \\
\hline
 & z_{n+1}z_n...z_1
\end{array}
$$

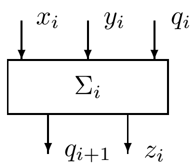
Nesunku apskaičiuoti z_i bei q_i reikšmes, kai x_i ir y_i yra duoti.

$$
\begin{cases}
z_i = x_i \oplus y_i \oplus q_i \\
q_{i+1} = (x_i \& y_i) \vee (x_i \& q_i) \vee (y_i \& q_i)
\end{cases}
$$

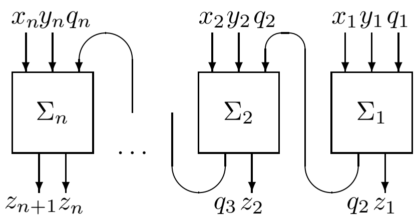
Kadangi pas mus nėra loginių elementų realizuojančių $\oplus$, tai tenka pirmosios lygties formulę išreikšti per $\neg$, $\&$, $\vee$.

$$
\begin{cases}
z_i = (\neg((x_i \& y_i) \vee (x_i \& q_i) \vee (y_i \& q_i)) \& (x_i \vee y_i \vee q_i)) \vee (x_i \& y_i \& q_i) \\
q_{i+1} = (x_i \& y_i) \vee (x_i \& q_i) \vee (y_i \& q_i)
\end{cases}
$$

Pažymėkime schemą, realizuojančią aprašytąją lygčių sistemą, kurioje trys įėjimai x_i, y_i, q_i ir du išėjimai z_i, q_{i+1}, simboliu Σ_i.



Dvejetainis sumatorius atrodys šitaip:



2.8 Pratimai

1. Sudaryti teisingumo lenteles:

a) $(p \to q)\&(\neg(\neg p \vee r) \to (q\&\neg r))$,

b) $((p\&q) \to (\neg p\&r))\&((q \to r) \vee (\neg p \to q))$,

c) $(p \to q) \to (q \to p)$.

2.Eliminuoti konstantas:

a) $((p \to t)\&(q \to k)) \vee (r\&t)$,

b) $((t \to p)\&(q\&k)) \to (r \to t)$,

c) $(p \vee k)\&((q \vee k) \to (p \to k))$.

3. Išreikšti disjunkcija ir neiginiu šias formules:

a) $p \to (q \to p)$,

b) $(p \to q) \vee (q \vee p)$,

c) $p \to (q \to (p\&q))$,

d) $(\neg p \to \neg q) \to (q \to p)$.

ir patikrinti ar jos tapačiai teisingos.

4. Naudojantis teisingumo lentelėmis, rasti tobulas normaliąsias disjunkcines formas:

a) $(p \to q)\&(q \to r)$,

b) $(p\&q) \to (q\&(\neg p \to r))$.

5. Naudojantis ekvivalenčiomis formulėmis, transformuoti į NDF:

a) $(p \vee q) \& \neg(p \rightarrow r)$,

b) $((p \& q) \rightarrow r) \& (\neg(p \& q) \rightarrow r)$.

6. Naudojantis ekvivalenčiomis formulėmis, transformuoti į NKF:

a) $(p \rightarrow q) \& (q \rightarrow (\neg p \rightarrow r))$,

b) $(p \vee r) \& (p \rightarrow (q \rightarrow \neg r))$.

7. Rasti formulėms atitinkančius Žegalkino polinomus:

a) $(p \vee q) \rightarrow (p \& q)$,

b) $p \rightarrow (q \rightarrow r)$,

c) $(p \& q) \rightarrow ((r \& \neg p) \rightarrow \neg q)$.