

3. Būla funkcijas un formulas

Nodaļas saturs

3.	Būla funkcijas un formulas	3-1
3.1.	Izteikumi.....	3-1
3.2.	Funkcijas	3-2
3.2.1.	Populārākās Būla funkcijas	3-2
3.2.2.	Viena un divu argumentu Būla funkciju kopsavilkums	3-2
3.3.	Formulas.....	3-3
3.3.1.	Formulas pieraksts un tās vērtību tabulas noskaidrošana.....	3-3
3.3.2.	Ekvivalentas formulas	3-5
3.3.3.	Substitūcija	3-5
3.4.	Disjunktīvā un konjunktīvā normālforma	3-5
3.4.1.	Definīcija un piemēri.....	3-5
3.4.2.	Pilnīga disjunktīvā normālforma	3-6
3.5.	Vingrinājumi	3-6

Nodaļā ievietotie attēli, programmu piemēri un tabulas

Tab. 3-1.	Būla funkciju AND, OR un NOT vērtību tabula	3-2
Tab. 3-2.	XOR, ekvivalences un implikācijas vērtību tabula	3-2
Tab. 3-3.	Viena argumenta Būla funkciju vērtību tabula	3-3
Tab. 3-4.	Divu argumentu Būla funkciju vērtību tabula	3-3

3.1. Izteikumi

Loģiskā programmēšana ir cēlusies uz matemātiskās loģikas ideju un principu pamata, tāpēc matemātiskās loģikas pamatzināšanas būtu ļoti vēlamas, lai labāk saprastu dažādu loģiskās programmēšanas valodu konstrukciju veidošanu un izpildi.

Viens no pamatjēdzieniem matemātiskajā loģikā ir izteikums.

Izteikums (*proposition*) ir matemātisks mainīgais vai konstrukcija, kam var būt divas patiesumvērtības: “patiess” (p, true, T, 1) vai “aplams” (a, false, F, 0).

Ikdienas dzīvē izteikums reprezentē kādu noteiktu atziņu, secinājumu vai spriedumu, piemēram, “spīd saule”, tomēr no matemātiskās loģikas manipulāciju viedokļa ir svarīgi tikai tas, ka šāds mainīgais spēj pieņemt jau pieminētās divas vērtības.

Šajā materiālā “patiess” parasti tiks apzīmēts ar 1, bet “aplams” ar 0.

No pamatzteikumiem ar loģisko operāciju palīdzību veido **saliktus izteikumus**, piemēram, “spīd saule un ir silts”, “ja spīd saule un līst lietus, tad ir redzama varavīksne”, “ja ir tumšs vai arī nāk miegs, tad ir nakts”.

Gramatikas formāla definīcija tiks apskatīta zemāk.

Gramatikas šajā mācību materiālā iekļautas tāpēc, ka to veidošana daudzos gadījumos atgādina Prolog programmas veidošanu un prasa līdzīgas iemaņas, tādējādi ilustrējot to, ka Prolog programmēšanas “greizā” domāšana (salīdzinot ar tradicionālo programmēšanu) ir pārstāvēts arī citās sfērās un ir uzskatāms par pietiekoši dabisku informācijas reprezentācijas veidu.

3.2. Funkcijas

3.2.1. Populārākās Būla funkcijas

Būla funkcija ir n argumentu funkcija $f=f(x_1, x_2, \dots, x_n)$ no mainīgajiem $x_1, x_2, \dots, x_n$, kuru vērtības ir 0 vai 1, un arī pašas funkcijas f vērtības ir 0 vai 1.

Trīs populārākās Būla funkcijas ir $\wedge$ (AND, &, “loģiskais un”), $\vee$ (OR, “loģiskais vai”), $\neg$ (NOT, “loģiskais nē”).

Tab. 3-1. Būla funkciju AND, OR un NOT vērtību tabula.

x	y	x AND y ($x \wedge y$)	x OR y ($x \vee y$)	NOT x ($\neg x, \bar{x}$)
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

AND un OR ir divargumentu funkcijas, bet NOT vienargumenta.

OR ir patiesa, ja kaut viens no argumentiem ir patiesi. AND ir patiesa tikai tad, ja abi argumenti ir patiesi. NOT atgriež vērtību, kas ir pretēja tās vienīgajam argumentam.

Starp populārākajām divargumentu Būla funkcijām ir arī “izslēdzošais un” (XOR, exclusive OR), $\rightarrow$ (implikācija) un $\equiv$ (ekvivalence).

Tab. 3-2. XOR, ekvivalences un implikācijas vērtību tabula.

x	y	x XOR y ($x \oplus y$)	$x \equiv y$	$x \rightarrow y$
0	0	0	1	1
0	1	1	0	1
1	0	1	0	0
1	1	0	1	1

XOR, atšķirībā no OR, ir patiesa tikai tad, ja ir patiesi tieši viens arguments (nevis abi reizē). Dabiskajās valodās lietotais saiklis “vai” atkarībā no situācijas tiek lietots vai nu parastā vai izslēdzošā vai nozīmē. Ekvivalence atgriež patiesu vērtību, ja abi argumenti ir vienādi. Implikācija, kas dabiskajā valodā atbilst slēdziena saiklim “ja, tad” vai “no x seko y ”, sākumā liekas visai dīvaina, jo sākumā ir grūti pieņemt, ka $0 \rightarrow 1 = 1$ (t.i., ir taisnība, ka no aplama, seko patiesi).

3.2.2. Viena un divu argumentu Būla funkciju kopsavilkums

Dažādo Būla funkciju skaits pie noteikta argumenta skaita ir viegli aprēķināms un ir 2^{2^n} .

Tādējādi viena argumenta Būla funkciju kopā ir 4, bet divu argumentu 16. No šīs formulas izriet arī, ka Būla funkciju bez neviena argumenta ir 2 (konstantes 0 un 1). Palielinoties n ,

dažādo Būla funkciju skaits ļoti strauji palielinās: 3 argumentu Būla funkciju ir 256, bet 4 argumentu – jau 65536 utt.

Tab. 3-3. Viena argumenta Būla funkciju vērtību tabula.

	x	0	1	
Nosaukums	Apzīmējums			Fiktīvie
nulle	0	0	1	x
identitāte	x	1	0	
noliegums	$\neg x, \bar{x}, x', \text{NOT } x$	1	0	
vieninieks	1	0	1	x

Šeit var ieviest jēdzienus – būtiskie un nebūtiskie mainīgie.

Būla funkcija f ir būtiski atkarīga no mainīgā x_i , ja eksistē tāds vērtību kopums $a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_n$, kur $f(a_1, \dots, a_{i-1}, 0, a_{i+1}, \dots, a_n) \neq f(a_1, \dots, a_{i-1}, 1, a_{i+1}, \dots, a_n)$.

Šādā gadījumā mainīgo x_i sauc par **būtisku** mainīgo, bet pretējā gadījumā par **nebūtisku** jeb **fiktīvu**. Tādējādi fiktīvs mainīgais ir tāds, kuram nav nekādas ietekmes uz funkcijas vērtību. Viena un divu argumentu Būla funkciju vērtību tabulās ir norādīti fiktīvie mainīgie katrai funkcijai.

Tab. 3-4. Divu argumentu Būla funkciju vērtību tabula.

	x	0	0	1	1	
	y	0	1	0	1	
Nosaukums	Apzīmējums					Fiktīvie
nulle	0	0	0	0	0	x,y
konjunkcija	$\cdot, \&, \wedge, \text{AND}$	0	0	0	1	
		0	0	1	0	
		0	0	1	1	y
		0	1	0	0	
		0	1	0	1	x
saskaitīšana pēc moduļa 2	$+, \oplus, \text{XOR}$	0	1	1	0	
disjunktija	$\vee, \text{OR}$	0	1	1	1	
Pīrsa bulta	$\downarrow$	1	0	0	0	
ekvivalence	$\sim, \equiv$	1	0	0	1	
		1	0	1	0	x
		1	0	1	1	
		1	1	0	0	y
implikācija	$\rightarrow, \Rightarrow, \supset$	1	1	0	1	
Šefera svītra	$ $	1	1	1	0	
vieninieks	1	1	1	1	1	x,y

3.3. Formulas

3.3.1. Formulas pieraksts un tās vērtību tabulas noskaidrošana

Formula ir konstrukcija, kas sastāv no loģiskajiem mainīgajiem un loģiskajām operācijām un reprezentē noteiktu Būla funkciju.

(Būla loģikas) formulas ir ērts veids, kā aprakstīt Būla funkcijas. Formulās parasti izmanto infikso pieraksta formu, kā arī ņem vērā prioritātes ($\neg, \wedge, \vee, \rightarrow$) un liekās iekavas tiek izlaistas.

Formulu veidošanai var tikt izmantoti dažādi operāciju komplekti, piemēram, $(\neg, \wedge, \vee, \rightarrow)$, $(\neg, \wedge, \vee)$ u.c. Teorētiski pietiek pat ar komplektiem $(\neg, \wedge)$ un $(\neg, \vee)$, bet tas ne vienmēr ir pietiekoši ērti.

Būla funkcijas, kuras atbilst formulā lietotajām operācijām sauc par **bāzes funkcijām**. Ja ir pieejamas vērtību tabulas bāzes funkcijām, tad var izrēķināt arī tās funkcijas vērtības, kuru realizē dotā formula.

Piemērs 1. $F_1 = x \wedge \neg y \vee \neg x \wedge y$

x	y	$\neg y$	$x \wedge \neg y$	$\neg x$	$\neg x \wedge y$	F_1
0	0	1	0	1	0	0
0	1	0	0	1	1	1
1	0	1	1	0	0	1
1	1	0	0	0	0	0

Kā redzams no rezultāta, formula F_1 realizē izslēdzošo “vai” (XOR).

Piemērs 2. $F_2 = \neg(\neg x \wedge \neg y \vee x \wedge y)$

x	y	$\neg x$	$\neg y$	$\neg x \wedge \neg y$	$x \wedge y$	$\neg x \wedge \neg y \vee x \wedge y$	F_2
0	0	1	1	1	0	1	0
0	1	1	0	0	0	0	1
1	0	0	1	0	0	0	1
1	1	0	0	0	1	1	0

Arī formula F_2 realizē izslēdzošo “vai” (XOR).

Piemērs 3. $F_3 = x \wedge \neg(\neg y \wedge \neg z) \vee \neg x \wedge y \wedge z$

x	y	z	$\neg y$	$\neg z$	$\neg y \wedge \neg z$	$\neg(\neg y \wedge \neg z)$	$x \wedge \neg(\neg y \wedge \neg z)$
0	0	0	1	1	1	0	0
0	0	1	1	0	0	1	0
0	1	0	0	1	0	1	0
0	1	1	0	0	0	1	0
1	0	0	1	1	1	0	0
1	0	1	1	0	0	1	1
1	1	0	0	1	0	1	1
1	1	1	0	0	0	1	1

$\neg x$	$\neg x \wedge y$	$\neg x \wedge y \wedge z$	F_3
1	0	0	0
1	0	0	0
1	1	0	0
1	1	1	1
0	0	0	0
0	0	0	1
0	0	0	1
0	0	0	1

Formula F_3 realizē mažoritāro funkciju $x\#y\#z$, kas atgriež patiesu vērtību, ja vairākums (šeit vismaz 2) argumenti ir patiesi.

3.3.2. Ekvivalentas formulas

Ekvivalentas ir tādas formulas, kuras realizē vienu un to pašu funkciju. Piemēram, iepriekšējā nodaļā (3.3.1) formulas F_1 un F_2 ir ekvivalentas, jo abas realizē XOR funkciju. Te arī ir redzams vienkāršākais veids, kā parādīt, ka divas formulas ir ekvivalentas: izrēķināt abu formulu vērtību tabulas un pārliedzināties, ka tās ir vienādas.

Tālāk parādītas pazīstamākās ekvivalentās formulas.

- | | | |
|-----|--|--|
| 1. | $x \vee x = x$ | $x \wedge x = x$ |
| 2. | $x \vee y = y \vee x$ | $x \wedge y = y \wedge x$ |
| 3. | $x \vee (y \vee z) = (x \vee y) \vee z$ | $x \wedge (y \wedge z) = (x \wedge y) \wedge z$ |
| 4. | $(x \wedge y) \vee x = x$ | $(x \vee y) \wedge x = x$ |
| 5. | $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$ | $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$ |
| 6. | $x \vee 1 = 1$ | $x \wedge 0 = 0$ |
| 7. | $x \vee 0 = x$ | $x \wedge 1 = x$ |
| 8. | $\neg\neg x = x$ | |
| 9. | $\neg(y \vee z) = \neg y \wedge \neg z$ | $\neg(y \wedge z) = \neg y \vee \neg z$ |
| 10. | $x \vee \neg x = 1$ | $x \wedge \neg x = 0$ |

Praktiski ļoti svarīga nozīme ir ekvivalencēm #9, jo tās parāda veidu, kā ar konjunkcijas ($\wedge$) palīdzību var izteikt disjunkciju ($\vee$) un otrādi. Šo formulu praktiski izmantojamie pārveidojumi būtu šādi:

$$y \vee z = \neg(\neg y \wedge \neg z)$$

$$y \wedge z = \neg(\neg y \vee \neg z)$$

3.3.3. Substitūcija

Substitūcija ir tāds formulas F pārveidojums, kad formulā F sastopamā apakšformula G tiek aizvietota ar citu apakšformulu H .

Substitūcijas rezultātā tiek iegūta jauna derīga formula.

To, ka formula F satur apakšformulu G , apzīmē šādi: $F(\dots G \dots)$. Ja formulā F **visas** apakšformulas G tiek aizvietotas ar H , tad to pieraksta šādi: $F(\dots G \dots)\{H//G\}$.

Piemērs 1. $x \vee \neg x \{y \wedge z // x\} = (y \wedge z) \vee \neg(y \wedge z)$

Piemērs 2. $x \vee y \vee z \{\neg x // y \vee z\} = x \vee \neg x$

Substitūcijas likums. Ja ekvivalentās formulās veic substitūciju, nomainot **visus** kādas apakšformulas gadījumus ar citiem, tad iegūtās formulas arī ir ekvivalentas:

$$\forall H (F_1(\dots G \dots) = F_2(\dots G \dots) \Rightarrow (F_1(\dots G \dots)\{H//G\} = F_2(\dots G \dots)\{H//G\}).$$

($\forall H$ šeit nozīmē: visiem H).

3.4. Disjunktīvā un konjunktīvā normālforma

3.4.1. Definīcija un piemēri

Lai padarītu Būla funkcijas pierakstu pārskatāmāku, tiek lietotas speciālas vienkāršākas formulas – normālformas. Izšķir divu veidu normālformas – disjunktīvās un konjunktīvās.

Par **disjunktīvu normālformu** sauc disjunkciju, kura sastāv no dažādu mainīgo vai to negāciju konjunkcijām (elementārajām konjunkcijām).

Disjunktīvās normālformas piemērs: $x \wedge y \vee y \wedge \neg z$

Par **konjunktīvu normālformu** sauc konjunkciju, kura sastāv no dažādu mainīgo vai to negāciju disjunkcijām (elementārajām disjunkcijām).

Konjunktīvās normālformas piemērs: $(\neg a \vee b) \wedge (\neg b \vee \neg c)$

Katrai funkcijai iespējamās daudzas disjunktīvās un konjunktīvās normālformas.

3.4.2. Pilnīga disjunktīvā normālforma

Disjunktīvās normālformas speciālgadījums ir pilnīgā disjunktīvā normālforma.

Par **pilnīgu disjunktīvu normālformu** sauc disjunktīvo normālformu, kura sastāv tikai no n -tā ranga elementārajām konjunkcijām (t.i., katrā no elementārajām konjunkcijām pārstāvēti visi mainīgie).

Formula F_1 nodaļā 3.3.1 ir pilnīga disjunktīva normālforma funkcijai XOR.

Pilnīgu disjunktīvu normālformu raksturo divas īpašības:

- katrai funkcijai eksistē viena vienīga šāda normālforma,
- šādu normālformu ir viegli iegūt no funkcijas vērtību tabulas.

Pilnīga disjunktīvā normālforma no attiecīgās funkcijas vērtību tabulas ir iegūstama šādā ceļā: atlasīt no tabulas visas mainīgo vērtību konfigurācijas (t.i., rindiņas), kuras dod funkcijas vērtību 1.

katra no šīm konfigurācijām būs viena elementārā konjunkcija, kur, ja argumenta vērtība ir 1, parādās attiecīgais mainīgais tiešā veidā, bet ja 0 – tad attiecīgā mainīgā negācija (sk. formulu F_1 nodaļā 3.3.1).

Līdzīgā veidā katrai funkcijai eksistē arī viena vienīga pilnīga konjunktīva normālforma.

3.5. Vingrinājumi

Vingrinājums #1.

Pierādīt, ka nodaļā 3.3.2 parādītās formulas 5a un 9b (a un b norāda attiecīgi formulu vienā vai otrā tabulas kolonnā) ir ekvivalences. Vingrinājums jāizpilda, izrēķinot attiecīgo formulu vērtību tabulas.

Vingrinājums #2.

Konstruēt pilnīgu disjunktīvu normālformu disjunkcijai $(x \vee y)$ un 3 argumentu mažoritārajai funkcijai (sk. piemēru #3 nodaļā 3.3.1).