ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ
КИЇВСЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ
ІМЕНІ ТАРАСА ШЕВЧЕНКА

В.В.Зубенко
О.В.Шишацька

МАТЕМАТИЧНА ЛОГІКА ТА ТЕОРІЯ АЛГОРИТМІВ
ПРИКЛАДИ І ЗАДАЧІ

Навчальний посібник

Київ 2023

УДК 510.51(075.8)
ББК 22.12я73
Ш55

Рецензенти
к.ф.-м.н., доцент Омельчук Л.Л. (доцент кафедри теорії та технології програмування Київського національного університету імені Тараса Шевченка)
к.ф.-м.н., доцент Шкільняк О.С. (доцент кафедри факультету комп'ютерних наук та кібернетики Київського національного університету імені Тараса Шевченка)

Рекомендовано до друку вченою радою факультету комп'ютерних наук та кібернетики (протокол № 9 від 21 березня 2023 року)

к.ф.-м.н., доц. Зубенко Віталій Володимирович
к.ф.-м.н., доц. Шишацька Олена Володимирівна

Математична логіка та теорія алгоритмів. Приклади і задачі / Електронне видання, - 2023. – 71 с.

Викладено матеріали до практичних занять з навчальної дисципліни «Матаматична логіка та теорія алгоритмів». В посібнику наведено умови практичних робіт та теоретичний матеріал для їх виконання, завдання для самостійної роботи та практичних занять, перелік контрольних запитань та базових задач до кожної теми.

Для студентів третього курсу факультету комп'ютерних наук та кібернетики Київського національного університету імені Тараса Шевченка, які навчаються за освітньо-професійною програмою «Прикладна математика» спеціальності 113 «Прикладна математика».

ЗМІСТ

1. МОДУЛЬ 1. ПРОПОЗИЦІЙНА ЛОГІКА

1.1. ПРАКТИЧНА РОБОТА 1. СИНТАКСИС ТА СЕМАНТИКА ПРОПОЗИЦІЙНОЇ ЛОГІКИ. ЛОГІЧНИЙ НАСЛІДОК В ПРОПОЗИЦІЙНОМУ ЧИСЛЕННІ

ВИКОРИСТАНО МАТЕРІАЛ:

1. ***Зубенко В.В., Шкільняк С.С. Основи математичної логіки: навчальний посібник -. К.: НУБіП України, 2020. - 102 с. (сторінка 14-23).***
2. ***Шкільняк С.С. Математична логіка. Приклади і задачі: Навчальний посібник. – К.: Видавничо-поліграфічний центр "Київський університет", 2007. – 145 с. (сторінка 17--21).***

Для точного дослідження предикатів та висловлень на пропозиційному рівні треба ввести *мову пропозиційної логіки* (ПЛ). Фактично така мова визначається семантичними моделями ПЛ. Множину базових пропозиційних композицій можна задавати різними способами, головне, щоб така множина визначала повний клас усіх пропозиційних композицій. Ми вибрали множину базових композицій $\{\neg, \vee\}$.

Алфавіт мови ПЛ складається із символів логічних зв'язок $\neg$ і $\vee$ та множини Ps пропозиційних символів (імен). Така множина Ps звичайно нескінченна.

Правильно побудовані вирази мови ПЛ називають *пропозиційними формулами* (ПФ). Для запису ПФ виберемо *префіксну* форму, коли символ операції передує аргументам. Множину всіх ПФ позначимо Fp.

Дамо індуктивне визначення пропозиційної формули.

1) кожний $A \in Ps$ є ПФ; такі ПФ назвемо *атомарними*;

2) якщо Φ та Ψ є ПФ, то $\neg\Phi$ та $\vee\Phi\Psi$ є ПФ.

Для бінарних операцій звичніше користуватися *інфіксною* формою, коли символ операції записується між аргументами, але тоді потрібні додаткові символи – дужки "(" і ")". Записи формул в інфіксній формі вважаємо скороченнями ПФ. Наприклад, вираз $\neg(\Phi\vee\Psi)$ – скорочення "справжньої" ПФ $\neg\vee\Phi\Psi$.

Пропозиційні композиції **&**, $\rightarrow$ та $\leftrightarrow$ можна виразити через $\neg$ та $\vee$. Тому вирази $\Phi\&\Psi$, $\Phi\rightarrow\Psi$ та $\Phi\leftrightarrow\Psi$ вважатимемо відповідно скороченнями ПФ $\neg\vee\neg\Phi\neg\Psi$, $\vee\neg\Phi\Psi$ та $\neg\vee\neg\vee\neg\Phi\Psi\neg\vee\neg\Psi\Phi$.

Для зменшення кількості дужок задамо в порядку спадання такий пріоритет символів логічних зв'язок: $\neg$, &, $\vee$, $\rightarrow$, $\leftrightarrow$. Введемо правило розставлення дужок справа наліво. Наприклад, $A\rightarrow B\rightarrow C\rightarrow\neg A\vee D$ означає $A\rightarrow(B\rightarrow(C\rightarrow((\neg A)\vee D)))$. Скорочення ПФ надалі також називатимемо ПФ.

Розглянемо традиційну інтерпретацію ПФ на множині висловлень за допомогою істиннісних оцінок.

Істиннісною оцінкою мови ПЛ назвемо довільне відображення $\tau: Ps\rightarrow\{T, F\}$. Для визначення значення усіх ПФ таке відображення $\tau: Ps\rightarrow\{T, F\}$ продовжимо до $\tau: Fp\rightarrow\{T, F\}$. Для цього покладемо

$\tau(\neg\Phi)=T \Leftrightarrow \tau(\Phi)=F$;

$\tau(\vee\Phi\Psi)=T \Leftrightarrow \tau(\Phi)=T$ або $\tau(\Psi)=T$.

Кожна ПФ з множиною пропозиційних імен X задає певну X-арну функцію на 2-елементній множині {*T*, *F*}, тобто *булеву функцію*. З іншого боку, кожна X-арна булева функція задається деякою ПФ, причому для фіксованої булевої функції множина таких ПФ нескінченна.

ПФ *тавтологія*, якщо вона має істиннісне значення *T* при кожній істиннісній оцінці мови ПЛ.

Отже, ПФ тавтологія, якщо вона істинна на кожному наборі значень її пропозиційних імен.

Суперечністю називають ПФ, якщо вона має істиннісне значення *F* при кожній істиннісній оцінці мови ПЛ.

Отже, ПФ суперечність, якщо вона хибна на кожному наборі значень її пропозиційних імен.

Кожна тавтологія задає тотальну булеву функцію, що приймає тільки значення 1, тобто константу 1.

Кожна суперечність задає тотальну булеву функцію, що приймає тільки значення 0, тобто константу 0.

Зрозуміло, що Φ тавтологія $\Leftrightarrow$ Φ суперечність.

Тавтології називають також *законами пропозиційної логіки.* Деякі тавтології мають власні назви.

Основоположні закони логіки виражають такі тавтології:

– закон тотожності: $P \leftrightarrow P$;

– закон виключеного третього (*tertium non datur*): $(\neg P) \vee P$;

– закон суперечливості (*lex contradictionis*): $\neg(P \& (\neg P))$.

До важливих законів пропозиційної логіки належать, зокрема, записані на мові ПЛ основні властивості пропозиційних композицій.

1) Закони комутативності для $\vee$, & та $\leftrightarrow$: $P \vee Q \leftrightarrow Q \vee P$; $P \& Q \leftrightarrow Q \& P$;

$(P \leftrightarrow Q) \leftrightarrow (Q \leftrightarrow P)$.

2) Закони асоціативності для $\vee$, & та $\leftrightarrow$: $(P \vee Q) \vee R \leftrightarrow P \vee (Q \vee R)$; $(P \& Q) \& R \leftrightarrow P \& (Q \& R)$;

$((P \leftrightarrow Q) \leftrightarrow R) \leftrightarrow (P \leftrightarrow (Q \leftrightarrow R))$.

3) Закони дистрибутивності для $\vee$ та &: $(P \vee Q) \& R \leftrightarrow (P \& R) \vee (Q \& R)$; $(P \& Q) \vee R \leftrightarrow (P \vee R) \& (Q \vee R)$.

4) Закон зняття подвійного заперечення: $\neg\neg P \leftrightarrow P$.

5) Закони ідемпотентності для $\vee$ та &: $P \leftrightarrow P \vee P$; $P \leftrightarrow P \& P$.

6) Закони де Моргана: $\neg(P \vee Q) \leftrightarrow (\neg P) \& (\neg Q)$; $\neg(P \& Q) \leftrightarrow (\neg P) \vee (\neg Q)$.

7) Закони поглинання: $P \to P \vee Q$; $P \& Q \to P$.

8) Закон контрапозиції: $(P \to Q) \leftrightarrow (\neg Q \to \neg P)$.

Задамо інтерпретацію ПФ на семантичних моделях ПЛ – пропозиційних композиційних системах.

При зафіксованій множині композицій $\boldsymbol{C}$ пропозиційна композиційна система $(A, Pr, \boldsymbol{C})$ однозначно визначається об'єктом вигляду (A, Pr). Такі об'єкти назвемо *абстрактними алгебраїчними системами* (ААС), їх теж будемо трактувати як семантичні моделі ПЛ.

Для завдання відображення інтерпретації $\boldsymbol{J} : Fp \to Pr$ спочатку визначимо відображення $\boldsymbol{I} : Ps \to Pr$, яке задає *базові* предикати системи (A, Pr). Тому інтерпретаціями мови ПЛ природно вважати об'єкти вигляду $((A, Pr), \boldsymbol{I})$. Такі об'єкти назвемо *ААС з доданими ПС* і будемо їх позначати у вигляді $\boldsymbol{A} = (A, \boldsymbol{I})$.

Відображення $\boldsymbol{J} : Fp \to Pr$ визначається за допомогою $\boldsymbol{I}$ так.

1. $\boldsymbol{J}(\mathrm{B})=\boldsymbol{I}(\mathrm{B})$ для кожного $\mathrm{B}\in Ps$.

2. Нехай $\boldsymbol{J}(\Phi)=P$. Тоді $\boldsymbol{J}(\neg\Phi)=\neg P$.

3. Нехай $\boldsymbol{J}(\Phi)=P$ та $\boldsymbol{J}(\Psi)=Q$. Тоді $\boldsymbol{J}(\vee\Phi\Psi)=P\vee Q$.

Предикат $\boldsymbol{J}(\Phi)$, який є значенням формули Φ при інтерпретації $\boldsymbol{A}=(A,\boldsymbol{I})$, позначатимемо Φ_A або $(\Phi)_A$.

Формула Φ *істинна при інтерпретації* $\boldsymbol{A}=(A,\boldsymbol{I})$, або $\boldsymbol{A}$*-істинна,* якщо Φ_A – істинний предикат. Цей факт позначатимемо $\boldsymbol{A}\models\Phi$.

Формула Φ *істинна,* якщо Φ істинна при кожній інтерпретації.

Зрозуміло, що пропозиційна формула Φ істинна $\Leftrightarrow$ Φ тавтологія. Отже, інтерпретація за допомогою істиннісних оцінок та інтерпретація на абстрактних алгебраїчних системах еквівалентні.

На множині ПФ введемо такі відношення:

– логічного (тавтологічного) наслідку $\models$;

– логічної (тавтологічної) еквівалентності $\sim_{\text{т}}$.

Формула Ψ є *логічним* (*тавтологічним*) *наслідком* формули Φ, що позначимо $\Phi\models\Psi$, якщо формула $\Phi\to\Psi$ – тавтологія.

Пропозиційні формули Φ та Ψ *логічно* (*тавтологічно*) *еквівалентні,* що позначимо $\Phi\sim_{\text{т}}\Psi$, якщо $\Phi\models\Psi$ та $\Psi\models\Phi$.

ПФ Ψ є *логічним* (*тавтологічним*) *наслідком множини ПФ* $\{\Phi_1,\ldots,\Phi_n\}$, що позначимо $\{\Phi_1,\ldots,\Phi_n\}\models\Psi$, якщо $\Phi_1\&\ldots\&\Phi_n\models\Psi$.

Замість $\varnothing\models\Phi$ писатимемо $\models\Phi$.

Вкажемо основні властивості відношень $\models$ та $\sim_{\text{т}}$.

1) відношення $\models$ рефлексивне і транзитивне;

2) відношення $\sim_{\text{т}}$ рефлексивне, транзитивне і симетричне.

3) Φ тавтологія $\Leftrightarrow$ $\models\Phi$;

4) $\Phi\sim_{\text{т}}\Psi\Leftrightarrow\models\Phi\leftrightarrow\Psi\Leftrightarrow\Phi\leftrightarrow\Psi$ тавтологія.

Приклад 1. Існують предикати P та Q такі, що P та $P\to Q$ частково істинні, але Q не істинний. Справді, нехай $A=\{0,1\}$. Покладемо $P(0)\uparrow$, $P(1)=T$, $Q(0)=F$, $Q(1)=T$. Тоді $(P\to Q)(0)\uparrow$ та $(P\to Q)(1)=T$. Отже, P та $P\to Q$ частково істинні, але Q не істинний.

Приклад 2. На АС $\boldsymbol{A}$ задамо предикат p_A як усюди невизначений, предикат q_A – як тотожно хибний. Тоді предикат $(p\to q)_A$ теж усюди невизначений. Отже, предикати p_A та $(p\to q)_A$ частково істинні. Таким чином, $\boldsymbol{A}\models p$ та $\boldsymbol{A}\models p\to q$, але $\boldsymbol{A}\not\models q$.

Для конкретних семантичних моделей можливо $\boldsymbol{A}\models\Phi$, $\boldsymbol{A}\models\Phi\to\Psi$ та $\boldsymbol{A}\not\models\Psi$.

Поняття логічного наслідку можна поширити на довільні множини формул.

Нехай $\Gamma\subseteq Fp$ та $\Delta\subseteq Fp$ – деякі множини формул.

Для спрощення запису замість $\{\Phi\}\cup\Gamma$ будемо звичайно писати Φ,Γ або Γ,Φ.

Скажемо, що *із* Γ *випливає* Δ, або Δ є *логічним наслідком* Γ, якщо для кожної істиннісної оцінки $\tau: Fp\to\{T,F\}$ із того, що $\tau(\Phi)=T$ для всіх $\Phi\in\Gamma$, випливає, що $\tau(\Psi)=T$ для деякої $\Psi\in\Delta$.

Те, що Δ є логічним наслідком Γ, будемо позначати $\Gamma \models \Delta$.

Звідси маємо: $\Gamma \not\models \Delta \Leftrightarrow$ існує істиннісна оцінка $\tau : Fp \to \{T, F\}$ така, що для всіх $\Phi \in \Gamma$ маємо $\tau(\Phi)=T$ та для всіх $\Psi \in \Delta$ маємо $\tau(\Psi)=F$.

Відношення логічного наслідку для множин формул рефлексивне, але нетранзитивне. Справді, очевидно $\Delta \models \Delta$, але із $\Gamma \models \Delta$ та $\Delta \models \Sigma$ не мусить випливати $\Gamma \models \Sigma$. Останнє засвідчує такий

Приклад 3. $\{\Phi \vee \Psi \vee \neg\Phi\} \models \{\Phi \vee \Psi, \Psi \vee \neg\Phi\}$ та $\{\Phi \vee \Psi, \Psi \vee \neg\Phi\} \models \Psi$, але $\{\Phi \vee \Psi \vee \neg\Phi\} \not\models \Psi$.

Приклад 4. Властивості відношення $\models$ на пропозиційному рівні.

G1) Якщо $\Gamma \cap \Delta \neq \varnothing$, то $\Gamma \models \Delta$.

G2) Нехай $\Gamma \models \Delta$ та $\Delta \subseteq \Sigma$. Тоді $\Gamma \models \Sigma$.

П1) $\neg\Phi, \Gamma \models \Delta \Leftrightarrow \Gamma \models \Delta, \Phi$.

П2) $\Gamma \models \Delta, \neg\Phi \Leftrightarrow \Phi, \Gamma \models \Delta$.

П3) $\Phi \vee \Psi, \Gamma \models \Delta \Leftrightarrow \Phi, \Gamma \models \Delta$ та $\Psi, \Gamma \models \Delta$.

П4) $\Gamma \models \Delta, \Phi \vee \Psi \Leftrightarrow \Gamma \models \Delta, \Phi, \Psi$.

П5) $\Phi \& \Psi, \Gamma \models \Delta \Leftrightarrow \Phi, \Psi, \Gamma \models \Delta$.

П6) $\Gamma \models \Delta, \Phi \& \Psi \Leftrightarrow \Gamma \models \Delta, \Phi$ та $\Gamma \models \Delta, \Psi$.

П7) $\Phi \to \Psi, \Gamma \models \Delta \Leftrightarrow \Gamma \models \Delta, \Phi$ та $\Psi, \Gamma \models \Delta$.

П8) $\Gamma \models \Delta, \Phi \to \Psi \Leftrightarrow \Phi, \Gamma \models \Delta, \Psi$.

П9) $\Phi \leftrightarrow \Psi, \Gamma \models \Delta \Leftrightarrow \Phi, \Psi, \Gamma \models \Delta$ та $\Gamma \models \Delta, \Phi, \Psi$.

П10) $\Gamma \models \Delta, \Phi \leftrightarrow \Psi \Leftrightarrow \Phi, \Gamma \models \Delta, \Psi$ та $\Psi, \Gamma \models \Delta, \Phi$.

Властивості П1–П4, в яких фігурують тільки базові композиції, назвемо базовими властивостями відношення $\models$ на пропозиційному рівні.

Теорема 1 (*семантичної еквівалентності*). *Нехай* Φ' *отримана із формули* Φ *заміною деяких входжень формул* $\Phi_1, ..., \Phi_n$ *на* $\Psi_1, ..., \Psi_n$ *відповідно. Якщо* $\Phi_1 \sim_\tau \Psi_1, ..., \Phi_n \sim_\tau \Psi_n$, *то* $\Phi \sim_\tau \Phi'$.

Теорема 2 (*заміни еквівалентних*). *Нехай* $\Phi \sim \Psi$. *Тоді*

$\Phi, \Gamma \models \Delta \Leftrightarrow \Psi, \Gamma \models \Delta$ *та* $\Gamma \models \Delta, \Phi \Leftrightarrow \Gamma \models \Delta, \Psi$.

ВПРАВИ

1. Доведіть:

1) $A \models A \vee B$;

2) $A \& B \models A$;

3) $\{A \vee B, \neg A \vee C\} \models B \vee C$;

4) $\{A \to B, C \to D\} \models A \vee C \to B \vee D$;

5) $\{A \to B, C \to D\} \models A \& C \to B \& D$.

4. Встановіть, чи вірно:

1) $\{A, A \to B\} \models B$?

2) $\{B, A \to B\} \models A$?

3) $\{\neg A, A\to B\} \models \neg B$?

4) $\{\neg B, A\to B\} \models \neg A$?

5) $\{A\to B, C\to D, A\vee D\} \models B\vee C$?

6) $\{A\vee B, B\to C, \neg A\vee\neg C\} \models \neg A\vee D$?

7) $\{A\vee B\to C\to D, \neg A\&\neg B\} \models C\&\neg D$?

8) $\{A\to C, B\to D, A\vee B\} \models D\&C$?

9) $\{A\to B, C\to D, \neg B\vee\neg D\} \models \neg A\vee\neg C$.

10) $\{A\to B, C\to D\} \models A\vee C\to B\&D$;

11) $\{A\to B, C\to D\} \models A\&C\to B\vee D$;

12) $A\&D\to B\vee C \models (A\to B)\vee(A\to C)\vee(D\to B)$.

5. В якому відношенні щодо $\models$ перебувають ПФ

$A\&B\to C\&D$, $A\vee B\to C\&D$, $A\&B\to C\vee D$, $A\vee B\to C\vee D$?

6. Знайдіть ПФ Ψ таку, щоб зазначена нижче ПФ була тавтологією:

1) $(\Psi\&A\to\neg B)\to((B\to\neg A)\to\Psi)$;

2) $((C\to\neg A\&B)\to\Psi)\to\Psi\&C\&(B\to A)$.

1.2. ПРАКТИЧНА РОБОТА 2. СЕКВЕНЦІЙНЕ ЧИСЛЕННЯ

ВИКОРИСТАНО МАТЕРІАЛ:

1. ***Зубенко В.В., Шкільняк С.С. Основи математичної логіки: навчальний посібник -. К.: НУБіП України, 2020. - 102 с. (сторінка 24-28).***
2. ***С.С. Шкільняк. Математична логіка. Приклади і задачі: Навчальний посібник. – К.: Видавничо-поліграфічний центр "Київський університет", 2007. – 145 с. (сторінка 25--30).***

Секвенційні числення пропозиційної логіки

Властивості відношення $\models$ дозволяють звести логічний наслідок складної формули (разом із множиною інших формул) до логічних наслідків простіших формул, що утворюють складнішу. Таким чином, питання про відношення логічного наслідку між двома множинами формул, в одну з яких входить складна формула, зводиться до питання про відношення логічного наслідку між двома множинами формул, в які вже входять компоненти складної формули. Це дуже важлива властивість *підформульності*.

Формально-аксіоматичні системи, які формалізують відношення логічного наслідку між двома множинами формул, називають *секвенційними численнями*. Основними об'єктами таких систем є секвенції, роль правил виведення грають секвенційні форми.

В традиційному, Генценівському варіанті, *секвенціями* називають об'єкти вигляду $\Gamma\to\Delta$, де Γ та Δ – множини формул, $\to$ – новий символ, що не входить до алфавіту мови логіки.

Така форма запису секвенцій цілком відповідає формі запису логічного наслідку для множин формул. Проте зручнішою є форма запису секвенцій за допомогою специфікованих формул, подібна до форми запису

семантичних таблиць Бета.

Кожну формулу секвенції відмітимо (специфікуємо) зліва одним з двох символів – $\vdash$ чи $\dashv$. Формулу зліва від $\to$ відмічаємо символом $\vdash$, формулу справа від $\to$ – символом $\dashv$. Така відмітка однозначно вказує на місце формули в секвенції – зліва чи справа від $\to$. Тому в записі секвенції символ $\to$ можна опустити.

Секвенцію, утворену з секвенції $\Gamma\to\Delta$ описаною вище відміткою формул, позначимо $\vdash\Gamma\dashv\Delta$. Не деталізуючи, секвенції відмічених формул також позначатимемо Σ.

Секвенційне числення будується так, що секвенція $\Gamma\to\Delta$ вивідна $\Leftrightarrow$ $\Gamma\models\Delta$.

Секвенція Σ *замкнена*, якщо існує формула Φ така, що $\vdash\Phi\in\Sigma$ та $\dashv\Phi\in\Sigma$.

Якщо $\Sigma=\vdash\Gamma\dashv\Delta$, це означає $\Gamma\cap\Delta\neq\varnothing$. Але з умови $\Gamma\cap\Delta\neq\varnothing$ випливає $\Gamma\models\Delta$, тому замкнені секвенції грають роль *аксіом*.

Семантичним властивостям П1–П4 зіставимо синтаксичні аналоги – *секвенційні форми*. Такі форми є правилами виведення секвенційних числень. Секвенційні форми звичайно записують у вигляді $\dfrac{\Sigma}{\Omega}$ або $\dfrac{\Sigma\quad\Lambda}{\Omega}$. Тут Σ, Λ, Ω – секвенції.

Секвенції над рискою називають засновками, секвенції під рискою – висновками. У нас засновки – це секвенції, зіставлені правим частинам відповідних семантичних властивостей, висновки – це секвенції, зіставлені їх лівим частинам.

Згідно з властивостями П1–П4 введемо такі базові секвенційні форми (зліва записуємо назву форми):

$\vdash\neg\ \dfrac{\dashv A,\Sigma}{\vdash\neg A,\Sigma}$ $\qquad$ $\dashv\neg\ \dfrac{\vdash A,\Sigma}{\dashv\neg A,\Sigma}$

$\vdash\vee\ \dfrac{\vdash A,\Sigma\qquad\vdash B,\Sigma}{\vdash A\vee B,\Sigma}$ $\qquad$ $\dashv\vee\ \dfrac{\dashv A,\dashv B,\Sigma}{\dashv A\vee B,\Sigma}$

Враховуючи властивості П1–П4, дістаємо:

Теорема 1. *Нехай* $\dfrac{\Sigma}{\Omega}$ та $\dfrac{\Sigma\quad\Lambda}{\Omega}$ *– секвенційні форми, причому* $\Sigma=\vdash\Lambda\dashv\mathbf{K}$, $\mathbf{Y}=\vdash\mathbf{X}\dashv\mathbf{Z}$ *та* $\Omega=\vdash\Gamma\dashv\Delta$. *Тоді:*

1) *якщо* $\Lambda\models\mathbf{K}$, *то* $\Gamma\models\Delta$;

2) *якщо* $\Lambda\models\mathbf{K}$ *та* $\mathbf{X}\models\mathbf{Z}$, *то* $\Gamma\models\Delta$. Враховуючи наявність похідних композицій & та $\to$, згідно з властивостями П5–П8 введемо такі похідні секвенційні форми:

$\vdash\to\ \dfrac{\dashv A,\Sigma\qquad\vdash B,\Sigma}{\vdash A\to B,\Sigma}$ $\qquad$ $\dashv\to\ \dfrac{\vdash A,\dashv B,\Sigma}{\dashv A\to B,\Sigma}$

$\vdash\&\ \dfrac{\vdash A,\vdash B,\Sigma}{\vdash A\,\&\,B,\Sigma}$ $\qquad$ $\dashv\&\ \dfrac{\dashv A,\Sigma\qquad\dashv B,\Sigma}{\dashv A\,\&\,B,\Sigma}$

Виведення у секвенційних численнях має вигляд дерева, вершинами якого є секвенції. Такі дерева називають *секвенційними*.

Формально поняття секвенційного дерева введемо індуктивно.

1) Секвенція Σ утворює *тривіальне* секвенційне дерево з єдиною вершиною Σ, яка є коренем дерева.

2) Нехай α – секвенційне дерево з коренем Σ, β – секвенційне дерево з коренем Υ, $\dfrac{\Sigma}{\Omega}$ та $\dfrac{\Sigma\ \Upsilon}{\Omega}$ – секвенційні форми. Тоді $\begin{array}{c}\alpha\\ |\\ \Omega\end{array}$ – секвенційне дерево з коренем Ω, $\begin{array}{c}\alpha\ \ \beta\\ \backslash\ /\\ \Omega\end{array}$ – секвенційне дерево з коренем Ω.

Тривіальне секвенційне дерево *замкнене*, якщо це замкнена секвенція. Нетривіальне секвенційне дерево *замкнене*, якщо кожний його лист (кінцева вершина, відмінна від кореня) – замкнена секвенція.

Секвенційне дерево з коренем Σ називають також секвенційним деревом секвенції Σ.

Секвенція Σ *вивідна*, або *має виведення*, якщо існує замкнене секвенційне дерево з коренем Σ. Таке дерево назвемо *виведенням* секвенції Σ.

Побудова дерева для секвенції Σ починається з кореня дерева, яке "росте" догори, в процесі побудови від складніших формул переходимо до простіших, тому секвенційні форми застосовуються знизу догори.

Процедуру побудови секвенційного дерева розіб'ємо на етапи. Кожне застосування секвенційної форми проводиться до скінченної множини *доступних* на даний момент формул. На початку кожного етапу виконується *крок доступу*: до списку доступних формул додаємо по одній формулі зі списків $\vdash$-формул та $\dashv$-формул. Якщо недоступних $\vdash$-формул чи $\dashv$-формул немає (відповідний список вичерпаний), то на подальших кроках доступу додаємо по одній формулі невичерпаного списку. На початку побудови дерева доступна лише пара перших формул списків.

Нехай виконано k етапів процедури.

На етапі k+1 перевіряємо, чи є кожен з листів дерева замкненою секвенцією. Якщо всі листи замкнені, то процедура завершена позитивно, маємо замкнене секвенційне дерево.

Якщо ні, то для кожного незамкненого листа ξ робимо наступний крок доступу, після чого добудовуємо скінченне піддерево з вершиною ξ таким чином.

Активізуємо всі доступні неатомарні формули ξ. По черзі до кожної активної формули застосовуємо відповідну секвенційну форму. Після виконання секвенційної форми формула пасивна, до пасивних та утворених на даному етапі формул секвенційні форми не застосовуються. Повтори формул усуваються.

При побудові секвенційного дерева можливі такі випадки:

1) Процедура завершена позитивно, маємо замкнене дерево.

2) Процедура завершена негативно, маємо скінченне або нескінченне незамкнене дерево. Тоді в дереві існує *незамкнений* шлях $\Sigma=\Sigma_1, \ldots, \Sigma_n, \ldots$, всі вершини якого – незамкнені секвенції. Кожна з формул секвенції Σ зустрінеться на цьому шляху і стане доступною, тому кожен із пропозиційних символів, що входять до складу формул секвенції Σ, зустрінеться на такому шляху як відмічена атомарна формула.

Теорема 2. *Нехай $\wp$ – незамкнений шлях у секвенційному дереві, $\boldsymbol{H}$ – множина всіх відмічених формул секвенцій цього шляху. Тоді існує істиннісна оцінка $\tau : Fp \to \{T, F\}$ така, що:*

1) *з умови $\vdash\Phi \in \boldsymbol{H}$ випливає $\tau(\Phi) = T$;*

2) *з умови $\dashv\Phi \in \boldsymbol{H}$ випливає $\tau(\Phi) = F$.*

Побудову секвенційних дерев зручніше вести зверху-вниз, тоді секвенційні форми перевертаються на 180°, їх висновки – над рискою, засновки – під рискою. Замкнені секвенції будемо відмічати справа символом ×.

Приклад 1. Для встановлення вірності $A\vee B\to C \models A\&B\to C$ будуємо виведення секвенції $\vdash A\vee B\to C, \dashv A\&B\to C$.

$\vdash A\vee B\to C, \dashv A\&B\to C$

$\dashv A\vee B, \dashv A\&B\to C$ $\qquad$ $\vdash C, \dashv A\&B\to C$

$\dashv A, \dashv B, \dashv A\&B\to C$ $\qquad$ $\vdash C, \vdash A\&B, \dashv C$ ×

$\dashv A, \dashv B, \vdash A\&B, \dashv C$

$\dashv A, \dashv B, \vdash A, \vdash B, \dashv C$ ×

Ми збудували замкнене секвенційне дерево, тому $A\vee B\to C \models A\&B\to C$.

Приклад 2. Для встановлення вірності $A\&B\to C \models A\vee B\to C$ будуємо виведення секвенції $\vdash A\&B\to C, \dashv A\vee B\to C$.

$\vdash A\&B\to C, \dashv A\vee B\to C$

$\dashv A\&B, \dashv A\vee B\to C$ $\qquad$ $\vdash C, \dashv A\vee B\to C$

$\dashv A, \dashv A\vee B\to C$ $\qquad$ $\dashv B, \dashv A\vee B\to C$ $\qquad$ $\vdash C, \vdash A\vee B, \dashv C$ ×

$\dashv A, \vdash A\vee B, \dashv C$ $\qquad$ $\dashv B, \vdash A\vee B, \dashv C$

$\dashv A, \vdash A, \dashv C$ × $\qquad$ $\dashv A, \vdash B, \dashv C$ $\qquad$ $\dashv B, \vdash A, \dashv C$ $\qquad$ $\dashv B, \vdash B, \dashv C$ ×

Отримали незамкнене дерево, тому невірно, що $A\&B\to C \models A\vee B\to C$. При цьому за двома незамкненими листами можна прочитати контрприклади: $\tau(A)=F, \tau(B)=T, \tau(C)=F$ та $\tau(A)=T, \tau(B)=F, \tau(C)=F$.

Вкажемо теореми коректності та повноти для пропозиційних секвенційних числень:

Теорема 3 (коректності). *Нехай секвенція* $\vdash\Gamma\dashv\Delta$ *вивідна. Тоді* $\Gamma \models \Delta$.

Теорема 4 (повноти). *Нехай* $\Gamma \models \Delta$. *Тоді секвенція* $\vdash\Gamma\dashv\Delta$ *вивідна.*

ВПРАВИ

1. Використовуючи властивості П5–П10 відношення $\models$ введіть похідні секвенційні форми $\vdash\to$, $\dashv\to$, $\vdash\&$, $\dashv\&$, $\vdash\leftrightarrow$, $\dashv\leftrightarrow$.

2. Доведіть теореми, сформульовані в розділі 2.3.

3. Побудова в пропозиційному секвенційному численні виведення формули Φ означає побудову виведення секвенції $\neg\Phi$. Збудуйте виведення чи доведіть його відсутність, вказавши контрприклад, для таких формул:

1) $(A\vee B\to C)\to(A\to C)$;

2) $(A\to B\&C)\to(A\to C)$;

3) $(A\to C)\to(B\to C)\to(A\to B\vee C)$;

4) $(A\to C)\to(B\to C)\to(C\to A\vee B)$;

5) $(A\to B)\to(A\to C)\to(A\&B\to C)$;

6) $(A\to B)\to(A\to C)\to(B\&C\to A)$;

7) $((A\to B)\to C)\to(A\vee C)$;

8) $(A\to B)\&(B\to C)\to(\neg A\to C)$;

9) $((A\to C)\to D)\&\neg D\to A\&\neg C$;

10) $(A\to B)\&(A\to C)\&(A\to D)\to\neg A$;

11) $(A\&B\to C)\to(A\to C)\vee(B\to C)$;

12) $((A\to B)\to(C\to D))\to(A\to C)\&(D\to B)$.

4. Використовуючи пропозиційне секвенційне числення, встановіть, чи вірно:

1) $\{A\vee B, \neg A\vee C\}\models B\vee C$?

2) $\{A\to B, C\to D\}\models A\vee C\to B\vee D$?

3) $\{A\to B, C\to D\}\models A\&C\to B\&D$?

4) $\{A\to B, C\to D, A\vee D\}\models B\vee C$?

5) $\{A\vee B, B\to C, \neg A\vee\neg C\}\models\neg A\vee D$?

6) $\{A\vee B\to C\to D, \neg A\&\neg B\}\models C\&\neg D$?

7) $\{A\to C, B\to D, A\vee B\}\models D\&C$?

8) $\{A\to B, C\to D, \neg B\vee\neg D\}\models\neg A\vee\neg C$?

9) $\{A\to B, C\to D\}\models A\vee C\to B\&D$?

10) $\{A\to B, C\to D\}\models A\&C\to B\vee D$?

11) $A\&D\to B\vee C\models(A\to B)\vee(A\to C)\vee(D\to B)$?

1.3. ПРАКТИЧНА РОБОТА 3. РЕЗОЛЮЦІЙНЕ ЧИСЛЕННЯ. ПРОПОЗИЦІЙНЕ ЧИСЛЕННЯ

ВИКОРИСТАНО МАТЕРІАЛ:

1. *Зубенко В.В., Шкільняк С.С. Основи математичної логіки: навчальний посібник -. К.: НУБіП України, 2020. - 102 с. (Пропозиційне числення - сторінка 31-26. Резолюційне пропозиційне числення – сторінка 29-31.)*
2. *С.С. Шкільняк. Математична логіка. Приклади і задачі: Навчальний посібник. – К.: Видавничо-поліграфічний центр "Київський університет", 2007. – 145 с. (Пропозиційне числення - сторінка 22-24. Резолюційне пропозиційне числення – сторінка 117--119.)*

Пропозиційне числення

Аксіоматичні системи гільбертівського типу базуються на понятті формальної системи. У випадку пропозиційної логіки вони називаються пропозиційними численнями, або численнями висловлень.

Під *пропозиційним численням* (ПЧ) розуміємо формальну систему (L, A, P), де L – мова ПЛ, A – множина *аксіом* ПЧ, P – множина *правил виведення* ПЧ.

Множина A задається єдиною схемою аксіом $\neg\Phi\vee\Phi$, тобто складається із всіх ПФ вигляду $\neg\Phi\vee\Phi$, які назвемо *пропозиційними аксіомами*.

Множина P складається з таких правил виведення:

П1) $\Phi \vdash \Psi\vee\Phi$ – правило розширення.

П2) $\Phi\vee\Phi \vdash \Phi$ – правило скорочення.

П3) $\Phi\vee(\Psi\vee\Xi) \vdash (\Phi\vee\Psi)\vee\Xi$ – правило асоціативності.

П4) $\Phi\vee\Psi, \neg\Phi\vee\Xi \vdash \Psi\vee\Xi$ – правило перетину.

Теоремою ПЧ називають ПФ, яка виводиться із пропозиційних аксіом за допомогою скінченної кількості застосувань правил виведення П1–П4.

Те, що ПФ Φ – теорема, позначатимемо $\vdash\Phi$.

Різні варіанти ПЧ можуть відрізнятися мовами та наборами аксіом і правил виведення.

Наведемо приклади виведень в ПЧ.

Приклад 1. Якщо $\vdash A\vee B$, то $\vdash B\vee A$.

$\vdash A\vee B$ за припущенням, $\vdash \neg\neg A\vee A$ як аксіома, тому $\vdash B\vee A$ за П2.

Приклад 2. Якщо $\vdash A$ та $\vdash A\to B$, то $\vdash B$.

$\vdash A$ за припущенням, $\vdash B\vee A$ за П1, звідки $\vdash A\vee B$. За припущенням $\vdash A\to B$, тобто $\vdash \neg\neg A\vee B$, тому $\vdash B\vee B$ за П4, звідки $\vdash B$ за П2.

Приклад 3. Якщо $\vdash A_1, ..., \vdash A_n$ та $\vdash A_1\to...\to A_n\to B$, то $\vdash B$.

Справді, застосуємо n раз твердження прикладу 2.2.2.

Твердження прикладів 2.2.1 та 2.2.2 можна інтерпретувати як нові, похідні правила виведення:

$A\vee B \vdash B\vee A$ – ***правило комутативності*** (скорочено ***ПК***);

$A, A\to B \vdash B$ – ***правило modus ponens*** (скорочено ***MP***).

Приклад 4. Якщо $\vdash (A\vee B)\vee C$, то $\vdash A\vee(B\vee C)$.

Із $\vdash (A\vee B)\vee C$ за ПК $\vdash C\vee(A\vee B)$, звідки $\vdash (C\vee A)\vee B$ за П3. Далі $\vdash B\vee(C\vee A)$ за ПК та $\vdash (B\vee C)\vee A$ за П3. Тепер за ПК $\vdash A\vee(B\vee C)$.

Твердження прикладу 2.2.4 та П3 в сукупності інтерпретуємо як нове, узагальнене правило виведення:

$\vdash (A\vee B)\vee C \Leftrightarrow \vdash A\vee(B\vee C)$ – ***повна асоціативність*** (скорочено ***АС***).

Розглянемо приклади виведень у пропозиційному численні.

Приклад 5. Якщо $\vdash A\vee B$, то $\vdash \neg\neg\neg A\vee B$.

Маємо $\vdash \neg\neg\neg A\vee\neg A$ (аксіома), тому $\vdash \neg\neg A\vee\neg\neg A$ за ПК. Звідси та з умови $\vdash A\vee B$ маємо $\vdash B\vee\neg\neg A$ за П4, тому $\vdash \neg\neg\neg A\vee B$ за ПК.

Приклад 6. $\vdash A\to B\to A$.

Із аксіоми $\vdash\neg\neg A\vee A$ за ПК $\vdash A\vee\neg A$. За П1 $\vdash\neg\neg B\vee(A\vee\neg A)$, за АС $\vdash(\neg B\vee A)\vee\neg A$, звідки $\vdash\neg\neg A\vee(\neg B\vee A)$ за ПК, тобто $\vdash A\to B\to A$.

Приклад 7. Якщо $\vdash A\vee B\vee A\vee C$, то $\vdash B\vee A\vee C$.

ПФ $B\vee A\vee C$ позначимо X. Із умови $\vdash A\vee X$ за ПК $\vdash X\vee A$. За П1 $\vdash C\vee(X\vee A)$, за АС $\vdash(C\vee X)\vee A$, звідки $\vdash A\vee(C\vee X)$ за ПК. За АС $\vdash(A\vee C)\vee X$, за П1 $\vdash B\vee((A\vee C)\vee X)$, за АС $\vdash(B\vee(A\vee C))\vee X$, тобто $\vdash X\vee X$. Тепер за П2 $\vdash X$, тобто $\vdash B\vee A\vee C$.

Коректність та повноту ПЧ гарантує

Теорема тавтології (ТТ) для ПЧ. *Множина теорем ЧВ співпадає з множиною тавтологій.*

Теорема тавтології ПЧ засвідчує адекватність семантичної та синтаксичної істинності, тобто повноту логічних засобів ПЧ.

Із ТТ безпосередньо випливає розв'язність ПЧ: множина теорем ПЧ алгоритмічно розв'язна відносно множини всіх ПФ.

На практиці найчастіше використовується такий

Наслідок ТТ. Якщо $\{A_1, ..., A_n\} \models A$ та $\vdash A_1,..., \vdash A_n$, то $\vdash A$.

Із наслідку ТТ, зокрема, випливають окремі наслідки:

1) $\vdash A\&B \Leftrightarrow \vdash A$ та $\vdash B$.

2) $\vdash A\to B \Leftrightarrow \vdash\neg\neg B\to\neg A$.

3) Якщо $\vdash A\to B$ та $\vdash B\to C$, то $\vdash A\to C$.

4) $\vdash A\leftrightarrow B \Leftrightarrow \vdash A\to B$ та $\vdash B\to A$.

5) Якщо $\vdash A\leftrightarrow B$, то ($\vdash A \Leftrightarrow \vdash B$).

Розглянемо приклади використання ТТ.

Приклад 8. Якщо $\vdash A\to B$ та $\vdash C\to D$, то $\vdash A\vee C\to B\vee D$.

Неважко переконатись, що $\{A\to B,\ C\to D\} \models A\vee C\to B\vee D$. Звідси і з умови $\vdash A\to B$ та $\vdash C\to D$ дістаємо $\vdash A\vee C\to B\vee D$ за ТТ.

Приклад 9. Якщо $\vdash A\vee C\to B\&D$, то $\vdash A\to B$ та $\vdash C\to D$.

Маємо $A\vee C\to B\&D \models (A\to B)\&(C\to D)$. Тепер скористуємось ТТ.

Приклад 10. Чи вірно: якщо $\vdash A\vee C\to B\vee D$, то $\vdash A\to B$ та $\vdash C\to D$?

Нехай із $\vdash A\vee C\to B\vee D$ випливає $\vdash A\to B$ та $\vdash C\to D$. Звідси необхідно $A\vee C\to B\vee D \models (A\to B)\&(C\to D)$. При $\tau(A)=\tau(C)=\tau(D)=T$ та $\tau(B)=F$ маємо $\tau(A\vee C\to B\vee D)=T$ та $\tau((A\to B)\&(C\to D))=F$.

Отже, з умови $\vdash A\vee C\to B\vee D$ не випливає $\vdash A\to B$ та $\vdash C\to D$.

ВПРАВИ

1. Доведіть в пропозиційному численні без використання ТТ:

1) $\vdash\neg\neg A\to A\to B$;

2) $\vdash B\to A\vee B$;

3) $\vdash A\to\neg A\to B$;

4) $\vdash(\neg A\to B)\vee\neg A$;

5) $\vdash A\vee A\to A$;

6) $\vdash(A\to B)\to(\neg B\to\neg A)$;

7) якщо $\vdash A\to C$, то $\vdash A\to B\vee C$;

8) якщо $\vdash A\vee B$, то $\vdash \neg A\to C\vee B$;

9) якщо $\vdash A\vee A\vee B$, то $\vdash A\vee B$;

10) якщо $\vdash A\vee B$, то $\vdash C\vee A\vee B$, $\vdash A\vee C\vee B$ та $\vdash A\vee B\vee C$;

11) якщо $\vdash A\vee B\vee C$, то $\vdash C\vee A\vee B$ та $\vdash A\vee C\vee B$;

12) якщо $\vdash A\vee B\vee C$, то $\vdash C\vee B\vee A$, $\vdash B\vee C\vee A$ та $\vdash B\vee A\vee C$;

13) якщо $\vdash A\to C$ та $\vdash B\to C$, то $\vdash A\vee B\to C$;

14) якщо $\vdash A\vee(B\to C)$ та $\vdash B$, то $\vdash A\vee C$;

15) якщо $\vdash A\to B\to C$ та $\vdash B$, то $\vdash A\to C$;

16) якщо $\vdash A\to B\to C$ та $\vdash A\to B$, то $\vdash A\to C$;

17) якщо $\vdash A\to B$, то $\vdash A\vee C\to B\vee C$.

2. Чи вірно: якщо ($\vdash A \Leftrightarrow \vdash B$), то $\vdash A\leftrightarrow B$?

3. Встановіть, які з наведених тверджень вірні:

1) якщо $\vdash A\&C\to B\&D$, то $\vdash A\to B$ та $\vdash C\to D$;

2) якщо $\vdash A\vee C\to B\&D$, то $\vdash A\to B$ та $\vdash C\to D$;

3) якщо $\vdash A\&C\to B\vee D$, то $\vdash A\to B$ та $\vdash C\to D$;

4) якщо $\vdash A\to B$ та $\vdash C\to D$, то $\vdash A\&C\to B\&D$;

5) якщо $\vdash A\to B$ та $\vdash C\to D$, то $\vdash A\&C\to B\vee D$;

6) якщо $\vdash A\to B$ та $\vdash C\to D$, то $\vdash A\vee C\to B\&D$.

4. Встановіть, які з наведених тверджень вірні:

1) a) якщо $\vdash(A\to B)\to C$, то $\vdash A\to C$;

   b) якщо $\vdash A\to C$, то $\vdash(A\to B)\to C$.

2) a) якщо $\vdash A\to B\&C$, то $\vdash A\to B$;

   b) якщо $\vdash A\to B$, то $\vdash A\to B\&C$.

3) a) якщо $\vdash A\&B\to C$, то $\vdash A\to C$;

   b) якщо $\vdash A\to C$, то $\vdash A\&B\to C$.

4) a) якщо $\vdash A\vee C\to B$, то $\vdash A\to B$;

   b) якщо $\vdash A\to B$, то $\vdash A\vee C\to B$.

2. МОДУЛЬ 2. ЛОГІКА ПЕРШОГО ПОРЯДКУ

2.1. ПРАКТИЧНА РОБОТА 4. МОДЕЛІ ТА МОВИ КЛАСИЧНИХ ЛОГІК ПЕРШОГО ПОРЯДКУ

ВИКОРИСТАНО МАТЕРІАЛ:

1. ***Зубенко В.В., Шкільняк С.С. Основи математичної логіки: навчальний посібник -. К.: НУБіП України, 2020. - 102 с. (сторінка 62-74).***
2. ***С.С. Шкільняк. Математична логіка. Приклади і задачі: Навчальний посібник. – К.: Видавничо-поліграфічний центр "Київський університет", 2007. – 145 с. (сторінка 113--116).***

Моделі та мови класичних логік першого порядку

Алгебраїчні системи. *Алгебраїчною системою* назвемо об'єкт вигляду $\boldsymbol{A} = (A, Fn^A \cup Pr^A)$, де A – непорожня множина, яку називають *носієм*, або *основою* АС, Fn^A та Pr^A – множини функцій та предикатів, заданих на A.

Нехай σ – довільна множина така, що існує тотальне однозначне відображення $\boldsymbol{I} : \sigma \to Fn^A \cup Pr^A$. Елементи множини σ трактуємо як *імена* деяких функцій та предикатів із $Fn^A \cup Pr^A$.

Такі імена називають функціональними символами (ФС) та предикатними символами (ПС), іменовані ними функції та предикати називають *базовими*.

Нехай Fs – множина функціональних символів, Ps – множина предикатних символів. Множину $\sigma = Fs \cup Ps$ називають *сигнатурою.*

АС із носієм A та сигнатурою $\sigma = Fs \cup Ps$ назвемо *АС з доданою сигнатурою*. Такі АС позначаємо у вигляді $\boldsymbol{A} = (A, \boldsymbol{I}, \sigma)$, або $\boldsymbol{A} = (A, \sigma)$, якщо $\boldsymbol{I}$ мається на увазі.

Для кожного $g \in Fs$ функцію $G \in Fn^A$ таку, що $\boldsymbol{I}(g)=G$, назвемо *значенням* ФС g при інтерпретації $\boldsymbol{I}$ на АС $\boldsymbol{A} = (A, Fn^A \cup Pr^A)$. Таку функцію позначатимемо g_A. Предикат $P \in Pr^A$ такий, що $\boldsymbol{I}(p)=P$, назвемо *значенням* ПС p при інтерпретації $\boldsymbol{I}$ на АС $\boldsymbol{A}$. Такий предикат позначатимемо p_A.

Якщо функція g_A є функція-константа на A, то ФС g називають *константним символом*.

Для класичних алгебраїчних систем [10] базові функції та предикати n-арні. Тому вважаємо, що з кожним ФС та ПС зв'язане натуральне число – арність такого символа. При цьому для кожного $h \in \sigma$ арність h_A рівна арності символу h.

АС $\boldsymbol{B} = (B, \sigma)$ назвемо *розширенням* АС $\boldsymbol{A} = (A, \sigma)$, якщо $A \subseteq B$ і для всіх $h \in \sigma$ та $a \in A$ маємо $h_A \subseteq h_B$. У цьому випадку АС $\boldsymbol{A}$ називають *підсистемою* АС $\boldsymbol{B}$, а $\boldsymbol{B}$ називають *надсистемою* АС $\boldsymbol{A}$. Цей факт позначатимемо $\boldsymbol{A} \subseteq \boldsymbol{B}$.

Нехай $\boldsymbol{A} = (A, \sigma)$. Множина $C \subseteq A$ утворює *підсистему* $\boldsymbol{C} = (C, \sigma)$ алгебраїчної системи $\boldsymbol{A} = (A, \sigma)$, якщо C замкнена відносно базових функцій f_A, де $f \in \sigma$.

Не для кожної $C \subseteq A$ можна говорити про підсистему (C, σ).

Приклад 1. Для АС (N, σ), де σ={+, =}, а символи + та = інтерпретуються як відповідні функції на N, множина непарних чисел $N_н \subseteq N$ незамкнена відносно +, тому $N_н$ не утворює підсистеми. В той же час множина парних чисел $N_п \subseteq N$ утворює власну підсистему ($N_п$, σ) системи (N, σ).

Нехай множини $A_1 \subseteq A$ та $A_2 \subseteq A$ замкнені відносно всіх базових функцій АС (A, σ). Тоді $A_1 \cap A_2$ теж замкнена відносно всіх базових функцій АС (A, σ), якщо тільки $A_1 \cap A_2 \neq \varnothing$. Отже, якщо ($A_1$, σ) та ($A_2$, σ) – підсистеми системи (A, σ), то або ($A_1 \cap A_2$, σ) – підсистема системи (A, σ), або $A_1 \cap A_2 = \varnothing$.

Підсистему ($A_1 \cap A_2$, σ) назвемо *перетином* підсистем (A_1, σ) та (A_2, σ).

Теорема 1. *Перетин M носіїв всіх підсистем системи* (A, σ) *або утворює підсистему* (M, σ), *або є* $\varnothing$.

Таку (M, σ) назвемо *найменшою підсистемою* системи (A, σ).

Зрозуміло, що якщо сигнатура σ містить константні символи, то АС (A, σ) має найменшу підсистему.

Нехай $\{A_\alpha\}_{\alpha \in J}$ – множина носіїв всіх підсистем системи $\boldsymbol{A}$ = (A, σ). Для довільної $B \subseteq A$ множина $C = \bigcap_{\alpha \in I} A_\alpha$, де I = $\{\alpha \in J \mid B \subseteq A_\alpha\}$, є найменшою множиною, замкненою відносно всіх базових функцій системи $\boldsymbol{A}$ = (A, σ). Така C визначає АС (C, σ), яку називають *підсистемою системи* (A, σ), *породженою множиною* B. Якщо при цьому C = A, то АС (A, σ) *породжується підмножиною* $B \subseteq A$.

Різні підмножини можуть породжувати одну і ту ж підсистему.

Приклад 2. Підсистема (Z^+, {1, +, =}) системи (N, {1, +, =}) породжується довільною підмножиною множини Z^+, яка містить 1. Найменшою такою підмножиною є {1}.

Приклад 3. Система (N, {+, =}) породжується множиною {0, 1}.

Приклад 4. Система (N, {0, 1, +, ×, =}) породжується множиною {0, 1}. Наявність константних символів 0 та 1 веде до того, що в кожній підсистемі такої системи носій містить 0 та 1, але тоді підсистема збігається з усією системою. Отже, така система власних підсистем не має.

Приклад 5. Система (Z^+, {+, =}) має підсистеми (kZ^+, {+, =}), де kZ^+={kx | $x \in Z^+$}, для довільних $k \in Z^+$.

Мови 1-го порядку. Засобами опису алгебраїчних систем є мови класичної логіки 1-го порядку, або просто *мови 1-го порядку*.

Алфавіт мови 1-го порядку складається із таких символів:

– множина V предметних імен (змінних);

– множина Fs функціональних символів заданої арності;

– множина Ps предикатних символів заданої арності;

– символи логічних операцій ¬, ∨ та ∃x.

У множині Fs може виділятися підмножина константних символів $Cn \subseteq Fs$. Символ рівності = завжди інтерпретуємо як предикат рівності, причому таку рівність трактуємо як тотожність.

Символи ¬, ∨, ∃, = та предметні імена назвемо *логічними* символами. Функціональні та предикатні символи, окрім =, назвемо *нелогічними* символами.

Множину σ = $Fs \cup Ps$ назвемо *сигнатурою* мови 1-го порядку.

Основними конструкціями мови 1-го порядку є *терми* та *формули*.

Терми використовують для позначення суб'єктів, формули – для запису тверджень про суб'єкти.

Індуктивне визначення *терма* таке:

1) кожне предметне ім'я та кожна константа є термом; такі терми назвемо *атомарними*;

2) якщо $t_1,..., t_n$ – терми, f – n-арний функціональний символ, то $ft_1...t_n$ – терм.

Атомарною формулою називається вираз виду $pt_1...t_n$, де p – n-арний предикатний символ, $t_1, ..., t_n$ – терми.

Індуктивне визначення *формули* таке:

1) кожна атомарна формула є формулою;

2) якщо Φ та Ψ – формули, то $\neg\Phi$ та $\vee\Phi\Psi$ – формули;

3) якщо Φ – формула, то $\exists x\Phi$ – формула.

Як і у мові ПЛ, вирази $\Phi\&\Psi$, $\Phi\to\Psi$ та $\Phi\leftrightarrow\Psi$ вважаємо скороченнями формул $\neg\vee\neg\Phi\neg\Psi$, $\vee\neg\Phi\Psi$ та $\neg\vee\neg\vee\neg\Phi\Psi\neg\vee\neg\Psi\Phi$. Користуємося також символом $\forall x$, вважаючи вираз $\forall x\Phi$ скороченням формули $\neg\exists x\neg\Phi$.

Для бінарних ФС та ПС і символів $\vee$, &, $\to$ та $\leftrightarrow$ звичайно застосовуємо інфіксну форму запису. Пріоритет символів логічних зв'язок вважаємо нижчим за пріоритет предикатних символів, а пріоритет предикатних символів нижчим за пріоритет функціональних символів. Використовуючи додаткові символи – кому "," і дужки "(" та ")", терми вигляду $ft_1...t_n$ записуємо $f(t_1...t_n)$, або t_1ft_2, якщо символ f бінарний. Те ж для атомарних формул. Атомарні формули вигляду $\neg{=}t_1\,t_2$ також записуємо $t_1{\neq}t_2$.

Скорочення термів та формул теж називатимемо термами та формулами.

Множини термів та формул позначаємо відповідно Tr та Fr.

Формули мови 1-го порядку сигнатури σ назвемо формулами 1-го порядку сигнатури σ.

Можна вказати два рівні відмінності мов 1-го порядку:

1) варіанти мови однієї сигнатури, що відрізняються наборами символів логічних операцій та способами запису термів і формул;

2) істотно різні мови, що відрізняються сигнатурами.

Мова 1-го порядку L' сигнатури σ' назвемо *розширенням* мови 1-го порядку L сигнатури σ, якщо $\sigma'\supseteq\sigma$.

У формулі вигляду $\exists x\Phi$ або $\forall x\Phi$ формулу Φ називають *областю дії* квантора $\exists x$ чи $\forall x$.

Вираз вигляду $\exists x$ або $\forall x$ називають *кванторним префіксом*.

Входження імені (змінної) x в формулу Φ *зв'язане*, якщо воно знаходиться в області дії деякого квантора по x, інакше таке входження x в Φ *вільне*.

Якщо існує вільне входження імені x в формулу Φ, то x – *вільне ім'я* (*вільна змінна*) формули Φ.

Формулу Φ із вільними іменами $x_1,.., x_n$ позначаємо $\Phi(x_1,.., x_n)$.

Формула *замкнена*, якщо вона не має вільних імен.

Терм, який не містить входжень предметних імен, називається *замкненим термом*. Зокрема, таким є кожний константний символ.

Наведемо кілька прикладів мов 1-го порядку.

Приклад 6. *Мова арифметики* L_{ar} визначається сигнатурою $\sigma_{ar} = \{0, 1, +, \times, =\}$, де 0 та 1 – константні символи, + та $\times$ – бінарні функціональні символи, = – бінарний предикатний символ.

Терм мови арифметики назвемо *арифметичним термом*.

Формулу мови арифметики назвемо *арифметичною формулою.*

Наприклад, 1+1 – замкнений арифметичний терм; $x\times(y+z)$ – арифметичний терм; $\exists z(x+z=y)$ – арифметична формула.

Приклад 7. Мова теорії множин L_{set} визначається сигнатурою $\sigma_{set}=\{\in, =\}$, де $\in$ та = – бінарні предикатні символи.

Наприклад, $z\in x$ – атомарна формула, $\forall z(z\in x\to z\in y)$ – формула, $\exists x\neg\exists y(y\in x)$ – замкнена формула мови L_{set}. Зауважимо, що останні дві формули відповідно означають "$x\subseteq y$" та "існує $\varnothing$".

Приклад 8. Мова теорії впорядкованих множин L_{ord} визначається сигнатурою $\sigma_{ord}=\{<, =\}$, де < та = – бінарні предикатні символи.

Наприклад, $x<y$ – атомарна формула, $z<x\to x<y\to z<y$ – формула, $\forall x\exists y(y<x)$ – замкнена формула мови L_{ord}.

Зв'язані імена в формулах можна замінювати іншими предметними іменами, але при цьому може виникнути *колізія* – ситуація, коли вільні імена стали зв'язаними.

Приклад 9. Із формули $\exists z(x+z=y)$ можна отримати формулу $\exists t(x+t=y)$, коли колізії нема, та формулу $\exists x(x+x=y)$, коли колізія змінила смисл формули.

Вільні входження предметних імен в формулу або терм можна замінювати термами. Позначимо $\Phi_{x_1,\ldots,x_n}[t_1, \ldots, t_n]$ формулу, отриману із формули Φ заміною всіх вільних входжень імен $x_1, \ldots, x_n$ на терми $t_1, \ldots, t_n$ відповідно.

Для термів аналогічно вводимо позначення $t_{x_1,\ldots,x_n}[t_1, \ldots, t_n]$

У загальному випадку формули $\Phi_{x,y}[a, b]$ та $(\Phi_x[a])_y[b]$ різні.

Наприклад, якщо Φ – це формула $x\in y$, то $\Phi_{x,y}[y, z]$ – це формула $y\in z$, $(\Phi_x[y])_y[z]$ – це формула $z\in z$.

При заміні вільних входжень предметних імен термами можливі *колізії*, коли вільне ім'я стає зв'язаним.

Приклад 10. Нехай Φ – це формула $\exists z(x+z=y)$. Тоді $\Phi_x[u]$ – це формула $\exists z(u+z=y)$; $\Phi_x[z]$ – це формула $\exists z(z+z=y)$; отже, маємо колізію.

Звідси маємо таке визначення.

Терм t *допустимий для заміни вільного імені* x *в формулі* Φ, якщо x не лежить в області дії ніякого квантора за деяким іменем, яке входить до складу терма t.

Інтерпретацією, або *моделлю* мови L сигнатури σ будемо називати АС з доданою сигнатурою вигляду $\boldsymbol{A} = (A, \boldsymbol{I}, \sigma)$.

Множину A називають *областю інтерпретації*.

Значення символів та виразів мови L задамо на $\boldsymbol{A}$ природним чином.

Предметні імена інтерпретуємо як імена елементів (змінні) множини A. Символи логічних операцій інтерпретуємо як відповідні логічні операції. Константні символи інтерпретуємо як конкретні елементи множини A, тобто як функції-константи на A. Предикатні та функціональні символи інтерпретуємо як предикати та функції відповідної арності на A, причому бінарний предикатний символ = завжди інтерпретуємо як предикат рівності на A.

Конкретна інтерпретація мови L на АС $\boldsymbol{A} = (A, \boldsymbol{I}, \sigma)$ визначається відображенням $\boldsymbol{I} : \sigma\to Fn^A\cup Pr^A$. Значення символів c, f, p позначаємо відповідно c_A, f_A, p_A: $\boldsymbol{I}(c)=c_A$, $\boldsymbol{I}(f)=f_A$, $\boldsymbol{I}(p)=p_A$.

Для інтерпретації термів і формул мови L задамо відображення $\boldsymbol{J} : Tr \cup Fr\to Fn^A \cup Pr^A$, яке індуктивно визначається за допомогою $\boldsymbol{I}$.

Для термів маємо:

– $\boldsymbol{J}(x) = \boldsymbol{'x}$;

– $\boldsymbol{J}(ft_1...t_n) = \boldsymbol{I}(f)(\boldsymbol{J}(t_1), ..., \boldsymbol{J}(t_n)) = f_A(\boldsymbol{J}(t_1), ..., \boldsymbol{J}(t_n))$.

Для атомарних формул маємо

– $\boldsymbol{J}(pt_1...t_n) = \boldsymbol{I}(p)(\boldsymbol{J}(t_1), ..., \boldsymbol{J}(t_n)) = p_A(\boldsymbol{J}(t_1), ..., \boldsymbol{J}(t_n))$.

Для формул маємо:

– нехай $\boldsymbol{J}(\Phi)=P$. Тоді $\boldsymbol{J}(\neg\Phi)=\neg P$, $\boldsymbol{J}(\exists x\Phi)=\exists \boldsymbol{x}P$.

– нехай $\boldsymbol{J}(\Phi)=P$ та $\boldsymbol{J}(\Psi)=Q$. Тоді $\boldsymbol{J}(\vee\Phi\Psi)=P\vee Q$.

Зауважимо, що тут неявно використовуємо операції суперпозиції: $f_A(g_1, ..., g_n)$ по суті означає $S^{1,...,n}(f_A, g_1, ..., g_n)$.

Кожний терм з вільними іменами $v_1, ..., v_n$ інтерпретуємо як $\{v_1, ..., v_n\}$-арну функцію на A, кожну формулу з вільними іменами $v_1, ..., v_n$ інтерпретуємо як $\{v_1, ..., v_n\}$-арний предикат на A. Зокрема, кожний замкнений терм інтерпретуємо як функцію-константу на A, кожну замкнену формулу – як предикат-константа на A.

Функцію, що є значенням терма t на АС $\boldsymbol{A} = (A, \boldsymbol{I}, \sigma)$, позначаємо t_A . Предикат, що є значенням формули Φ на АС $\boldsymbol{A} = (A, \boldsymbol{I}, \sigma)$, позначаємо Φ_A . Це означає, що $\boldsymbol{J}(t)=t_A$, $\boldsymbol{J}(\Phi)=\Phi_A$.

Формула Φ *істинна при інтерпретації* $\boldsymbol{A}$, або *істинна на* $\boldsymbol{A}$, або $\boldsymbol{A}$*-істинна*, якщо предикат Φ_A є істинним.

Це означає: X-арний предикат Φ_A такий, що $\Phi_A(d)=T$ для всіх $d\in A^X$.

Те, що формула Φ істинна на АС $\boldsymbol{A}$, позначаємо $\boldsymbol{A} \models \Phi$.

Формула Φ *всюди істинна*, якщо вона істинна при кожній інтерпретації.

Те, що Φ всюди істинна, позначимо $\models\Phi$.

Формула Φ *виконувана при інтерпретації* $\boldsymbol{A}$, або *виконувана на АС* $\boldsymbol{A}$, або $\boldsymbol{A}$*-виконувана*, якщо предикат Φ_A є виконуваним.

Це означає: X-арний предикат Φ_A такий, що $\Phi_A(d)=T$ для деякого $d\in A^X$.

Формула Φ *виконувана*, якщо Φ виконувана при деякій інтерпретації.

Приклад 11. Формула $x=x$ всюди істинна.

Приклад 12. Формула $\forall x\forall y(x=y)$ істинна на всіх 1-елементних АС і тільки на них; формула $\neg\forall x\forall y(x=y)$ істинна на всіх k-елементних АС, де $k>1$, і тільки на них.

Замиканням формули Φ з вільними іменами $x_1, ..., x_n$ назвемо замкнену формулу $\forall x_1...\forall x_n\Phi$, яку звичайно позначатимемо $\overline{\Phi}$.

Із визначень випливає *семантична теорема замикання:*

Теорема 1. *Для кожних АС* $\boldsymbol{A}$ *та формули* Φ $\boldsymbol{A} \models \Phi \Leftrightarrow \boldsymbol{A} \models \overline{\Phi}$.

Приклад 13. Кожна формула вигляду $\Phi_x[t]\to\exists x\Phi$ всюди істинна.

Нехай X – множина вільних імен формули $\Phi_x[t]\to\exists x\Phi$. Припустимо супротивне: існує $\boldsymbol{A} = (A, \sigma)$ така, що $\boldsymbol{A} \not\models \Phi_x[t]\to\exists x\Phi$. Тоді існує $d\in A^X$ таке, що $(\Phi_x[t]\to\exists x\Phi)_A(d)=F$, звідки $(\Phi_x[t])_A(d)=T$ та $(\exists x\Phi)_A(d)=F$. Нехай $t_A(d)=b\in A$; в силу $(\Phi_x[t])_A(d) = T$ тоді $\Phi_A(d\nabla x\mapsto b) = T$. Але $(\exists x\Phi)_A(d)=F$, тому $\Phi_A(d\nabla x\mapsto a)=F$ для всіх $a\in A$, зокрема, $\Phi_A(d\nabla x\mapsto b)=F$. Маємо суперечність.

Окремим випадком всюди істинних формул є *тавтології*, тобто формули, які мають структуру тавтологій мови ПЛ.

Формула *пропозиційно нерозкладна,* якщо вона атомарна або має вигляд $\exists x\Phi$.

Нехай Fr_0 – множина всіх пропозиційно нерозкладних формул мови *L*.

Істиннісна оцінка мови *L* – це довільне відображення $\tau : Fr_0 \to \{T, F\}$.

Продовжимо τ до відображення $\tau : Fr \to \{T, F\}$ таким чином:

– $\tau(\neg\Phi)=T \Leftrightarrow \tau(\Phi)=F$;

– $\tau(\vee\Phi\Psi)=T \Leftrightarrow \tau(\Phi)=T$ або $\tau(\Psi)=T$.

Формула Φ мови *L* *тавтологія*, якщо для кожної істиннісної оцінки τ мови *L* маємо $\tau(\Phi)=T$.

Кожна тавтологія є всюди істинною формулою, але зворотне невірне. Наприклад, всюди істинна формула вигляду $x=x$ – не тавтологія.

На множині формул введемо відношення тавтологічного наслідку $\models$, логічного наслідку $|=$, слабкого логічного наслідку $||=$, тавтологічної еквівалентності $\sim_т$ та логічної еквівалентності $\sim$.

Формула Ψ є *тавтологічним наслідком* формули Φ, що позначатимемо $\Phi \models \Psi$, якщо формула $\Phi \to \Psi$ – тавтологія.

Формули Φ та Ψ *тавтологічно еквівалентні,* що позначатимемо $\Phi \sim_т \Psi$, якщо $\Phi \models \Psi$ та $\Psi \models \Phi$.

Формула Ψ є *логічним наслідком* формули Φ, що позначатимемо $\Phi|=\Psi$, якщо формула $\Phi \to \Psi$ всюди істинна.

Формули Φ та Ψ *логічно еквівалентні,* що позначатимемо $\Phi \sim \Psi$, якщо $\Phi|=\Psi$ та $\Psi|=\Phi$.

Зрозуміло, що $\Phi \sim \Psi \Leftrightarrow$ формули $\Phi \to \Psi$ та $\Psi \to \Phi$ всюди істинні.

Формула Ψ є *слабким логічним наслідком* формули Φ, що позначаємо $\Phi||=\Psi$, якщо для кожної інтерпретації $\boldsymbol{A}$ із умови $\boldsymbol{A} |=\Phi$ випливає $\boldsymbol{A} |=\Psi$.

Формула Ψ є *логічним наслідком* множини формул $\{\Phi_1,...,\Phi_n\}$, що позначатимемо $\{\Phi_1,...,\Phi_n\}|=\Psi$, якщо $\Phi_1\&...\&\Phi_n |=\Psi$.

Аналогічно визначаємо $\{\Phi_1,...,\Phi_n\} \models \Psi$ та $\{\Phi_1,...,\Phi_n\}||=\Psi$.

Замість $\varnothing \models \Phi$, $\varnothing|=\Phi$ та $\varnothing||=\Phi$ пишемо відповідно $\models \Phi$, $|=\Phi$ та $||=\Phi$.

Вкажемо основні властивості для $\models$, $\sim_т$, $|=$, $||=$ та $\sim$:

1) Φ тавтологія $\Leftrightarrow \models \Phi$;

2) Φ всюди істинна $\Leftrightarrow |=\Phi \Leftrightarrow ||=\Phi$;

3) якщо $\Phi \models \Psi$, то $\Phi|=\Psi$; але не завжди із $\Phi|=\Psi$ випливає $\Phi \models \Psi$;

4) якщо $\Phi|=\Psi$, то $\Phi||=\Psi$; але не завжди із $\Phi||=\Phi$ випливає $\Phi|=\Psi$;

5) $\Phi \sim_т \Psi \Leftrightarrow \Phi \leftrightarrow \Psi$ тавтологія

6) $\Phi \sim \Psi \Leftrightarrow |=\Phi \leftrightarrow \Psi$;

7) відношення $\models$, $|=$ та $||=$ рефлексивні і транзитивні;

8) відношення $\sim_т$ та $\sim$ рефлексивні, транзитивні і симетричні.

Враховуючи 2), той факт, що Φ всюди істинна, позначаємо $|=\Phi$.

Для 3) та 4) маємо такі *контрприклади*:

Приклад 14. $\exists x\exists y(x=y)|=\exists y\exists x(x=y)$, але невірно $\exists x\exists y(x=y) \models \exists y\exists x(x=y)$.

Приклад 15. $(x=0)\|\!=\forall x(x=0)$ але невірно $(x=0)\models\forall x(x=0)$.

Справді, за теоремою замикання $(x=0)\|\!=\forall x(x=0)$. Але $(x=0)_N(0)=T$ та $(\forall x(x=0))_N=F$, тому $(x=0\to\forall x(x=0))_N(0)=F$, звідки $(x=0)\not\models\forall x(x=0)$.

Приклад 16. Якщо x не вільне в Ψ, то $\Phi\to\Psi\|\!=\exists x\Phi\to\Psi$.

Нехай X – множина вільних імен формули $\Phi\to\Psi$. Припустимо супротивне: існує $\boldsymbol{A}=(A,\sigma)$ така, що $\boldsymbol{A}\models\Phi\to\Psi$ та $\boldsymbol{A}\not\models\exists x\Phi\to\Psi$. Тоді існує $d\in A^X$ таке, що $(\exists x\Phi\to\Psi)_A(d)=F$, звідки $(\exists x\Phi)_A(d)=T$ та $\Psi_A(d)=F$. Згідно $(\exists x\Phi)_A(d)=T$ маємо $\Phi_A(d\nabla x\mapsto b)=T$ для деякого $b\in A$. Але x не вільне в Ψ, тому $\Psi_A(d\nabla x\mapsto b)=\Psi_A(d)=F$. Звідси $(\Phi\to\Psi)_A(d\nabla x\mapsto b)=F$, що суперечить $\boldsymbol{A}\models\Phi\to\Psi$.

Формула Φ мови L *k-істинна*, якщо $\boldsymbol{A}\models\Phi$ для кожної k-елементної інтерпретації $\boldsymbol{A}$ мови L.

Формула Φ *скінченно-істинна*, якщо Φ k-істинна для кожного $k>0$.

Отже, скінченно-істинна формула є істинною при кожній скінченній інтерпретації.

Приклад 17. Формула $\exists x_1\ldots\exists x_k((x_1\neq x_2)\&\ldots\&(x_1\neq x_k)\&(x_2\neq x_3)\&\ldots\&(x_{k-1}\neq x_k))$, яку позначимо E_k, стверджує, що існує $\geq k$ різних елементів області інтерпретації. Отже, E_k є n-істинною для всіх $n\geq k$.

Приклад 18. Формула $\exists x_1\exists x_2\ldots\exists x_k\forall y((y=x_1)\vee\ldots\vee(y=x_k))$, яку позначимо G_k, стверджує, що існує $\leq k$ різних елементів області інтерпретації. Отже, G_k є n-істинною для всіх $1\leq n\leq k$.

Приклад 19. Формула $E_k\&G_k$ k-істинна, причому така $E_k\&G_k$ не є n-істинною для кожного $n\neq k$.

ВПРАВИ

Доведіть, що система $(Z, \{+, -, \times, =\})$ має найменшу підсистему $(\{0\}, \{+, -, \times, =\})$.

3. Доведіть, що система $(N, \{\times, =\})$ не породжується жодною скінченою підмножиною $B\subseteq N$.

4. Чи має найменшу підсистему система $(Z^+, \{+, =\})$?

5. Наведіть приклади "природних" 2-елементних інтерпретацій L_{ar} .

6. Вкажіть формули L_{set}, які означають:

1) "$X\subset Y$";

2) "$X=Y\cup Z$";

3) "$X=(Y\cap Z)\setminus S$";

4) "$X=Y\setminus(Z\cup S)$";

5) "$X=Y\cap(Z\setminus S)$";

6) "$X\cap Y=Z\setminus S$";

7) "$X=2^Y$";

8) "$Z\cup S=2^{X\setminus Y}$".

7. Доведіть чи спростуйте такі твердження:

1) $\models\forall x\exists y A\to\exists y\forall x A$;

2) $\models\exists y\forall x A\to\forall x\exists y A$;

3) $\models\forall x A\vee\forall x B\to\forall x(A\vee B)$;

4) $\models\forall x(A\vee B)\to\forall x A\vee\forall x B$;

5) $\models \exists xA \vee \exists xB \to \exists x(A \vee B)$;

6) $\models \exists x(A \vee B) \to \exists xA \vee \exists xB$;

7) $\models \forall xA \& \forall xB \to \forall x(A \& B)$;

8) $\models \forall x(A \& B) \to \forall xA \& \forall xB$;

9) $\models \exists xA \& \exists xB \to \exists x(A \& B)$;

10) $\models \exists x(A \& B) \to \exists xA \& \exists xB$;

11) $\models \exists xA \vee B \to A \vee \exists xB$;

12) $\models A \& \forall xB \to \forall xA \& B$;

13) $\models \forall xA \vee \exists xB \to \exists xA \vee \forall xB$;

14) $\models \exists xA \vee \forall xB \to \forall xA \vee \exists xB$;

15) $\models (A \to B) \to (\exists xA \to \exists xB)$;

16) $\models (A \to B) \to (\forall xA \to \forall xB)$.

8. Чи вірно:

1) a) $\models \exists x(P \& Q) \to \exists xP \& Q$?

b) $\models \exists xP \& Q \to \exists x(P \& Q)$?

c) $\forall xP \to \exists xQ \mid\mid \models \forall xP \to Q$?

2) a) $P \to Q \models \forall xP \to \forall xQ$?

b) $\forall xP \to \forall xQ \models P \to Q$?

c) $\forall xP \to \forall xQ \mid\mid \models \exists xP \to \exists xQ$?

3) a) $\forall xP \to Q \models \forall x(P \to Q)$?

b) $\exists x(P \to Q) \models \exists xP \to Q$?

c) $\forall xP \to \exists xQ \mid\mid \models P \to \exists xQ$?

4) a) $\models \forall x(P \vee Q) \to \forall xP \vee Q$?

b) $\models \forall xP \vee Q \to \forall x(P \vee Q)$?

c) $\exists xP \to \exists xQ \mid\mid \models \forall xP \to \forall xQ$?

9. В якому відношенні щодо $\models$ та $\mid\mid\models$ формули вигляду:

1) $A \to B$ та $\exists xA \to \exists xB$?

2) $A \to B$ та $\forall xA \to \forall xB$?

3) $\forall x(A \vee B)$ та $\forall xA \vee \forall xB$?

4) $\exists xA \& \exists xB$ та $\exists x(A \& B)$?

5) $\forall x(A \vee B)$ та $\forall xA \vee B$?

6) $\exists xA \vee B$ та $\exists x(A \vee B)$?

7) $\forall x(A \& B)$ та $\forall xA \& B$?

8) $\exists xA \& B$ та $\exists x(A \& B)$?

9) $\exists x(A \to B)$ та $\exists xA \to B$?

10) $\forall x(A \to B)$ та $\forall xA \to B$?

11) $\forall x A \to \exists x B$ та $A \to B$?

12) $\exists x A \to \forall x B$ та $A \to B$?

13) $A \vee B$ та $\exists x A \vee \forall x B$?

14) $A \& B$ та $\exists x A \& \forall x B$?

2.2. ПРАКТИЧНА РОБОТА 5. ВИРАЗНІСТЬ В АРИФМЕТИЧНИХ СИСТЕМАХ. ГОМОМОРФІЗМИ АЛГЕБРАЇЧНИХ СИСТЕМ. МЕТОД АВТОМОРФІЗМІВ

ВИКОРИСТАНО МАТЕРІАЛ:

1. ***Зубенко В.В., Шкільняк С.С. Основи математичної логіки: навчальний посібник -. К.: НУБіП України, 2020. - 102 с. (сторінка 49-52).***
2. ***С.С. Шкільняк. Математична логіка. Приклади і задачі: Навчальний посібник. – К.: Видавничо-поліграфічний центр "Київський університет", 2007. – 145 с. (сторінка 48-61).***

Виразність в АС. Арифметичні предикати, множини, функції

Нехай $\boldsymbol{A} = (A, \boldsymbol{I}, \sigma)$ – деяка АС. Предикат P на A *виразний формулою* Φ сигнатури σ, якщо P – суть предикат Φ_A.

Предикат P на A *виразний в АС* $\boldsymbol{A} = (A, \boldsymbol{I}, \sigma)$, якщо P виразний деякою формулою Φ сигнатури σ.

Інакше кажучи, предикат P на A виразний в АС $\boldsymbol{A} = (A, \boldsymbol{I}, \sigma)$, якщо існує така формула Φ сигнатури σ, що P – суть предикат Φ_A.

Множина, що є областю істинності предикату, виразного в АС $\boldsymbol{A}$, називається *виразною в АС* $\boldsymbol{A}$ *множиною*.

Функція, графік якої – виразна в АС $\boldsymbol{A}$ множина, називається *виразною в* АС $\boldsymbol{A}$ *функцією*.

Приклад 1. Предикат "$x=0$" в АС $(N, \{\times, =\})$, $(Q, \{\times, =\})$, $(R, \{\times, =\})$ виражається формулою $\forall y(x \times y = x)$.

Приклад 2. Предикат "$x=1$" в АС $(N, \{\times, =\})$, $(Z, \{\times, =\})$, $(R, \{\times, =\})$ виражається формулою $\forall y(x \times y = y)$.

Приклад 3. Предикат "$x=0$" в АС $(N, \{+, =\})$, $(Z, \{+, =\})$, $R, \{+, =\})$ виражається формулою $x+x=x$.

Приклад 4. Предикат "$x=1$" в АС $(N, \{+, =\})$ виражається формулою $\forall u \forall v(x=u+v \to u=u+u \vee v=v+v)$ & $\neg x=x+x$.

Приклад 5. Предикат "$y=x+4$" в АС $(R, \{y=x+2, =\})$ виражається формулою $\exists z(y=z+2 \& z=x+2)$.

Приклад 6. Предикат "$|x-y|=2$" в АС $(Z, \{|x-y|=1, =\})$ виражається формулою $\exists z(|x-z|=1 \& |z-y|=1 \& \neg x=y$.

Приклад 7. Предикат "$|x-y|=3$" в АС $(Q, \{y=x+3, =\})$ виражається формулою $y=x+3 \vee x=y+3$.

Приклад 8. Предикат "$z=x+1$" виражається в АС $(Z, \{< =\})$ формулою $(x<z) \& \neg \exists v(x<v \& v<z)$.

Множину натуральних чисел N з виділеними константами 0 та 1, визначеними на N стандартними бінарними функціями додавання + і множення $\times$ та стандартним предикатом рівності назвемо *стандартною інтерпретацією*, або *стандартною моделлю* мови арифметики.

Інакше кажучи, стандартна інтерпретація L_{ar} – це АС $\boldsymbol{N} = (N, \sigma_{ar})$.

Арифметична формула, яка істинна на $\boldsymbol{N}$, називається *істинною арифметичною формулою* (ІАФ).

Кожна всюди істинна арифметична формула є ІАФ, але не кожна ІАФ всюди істинна. Наприклад, формула $\neg\exists x(x+1=0)$ є ІАФ, але вона не істинна на $\boldsymbol{Z} = (Z, \sigma_{ar})$ та на $\boldsymbol{R} = (R, \sigma_{ar})$.

Предикати, множини та функції, виразні в $\boldsymbol{N} = (N, \sigma_{ar})$, назвемо *арифметичними.*

Отже, функція f арифметична, якщо її графік Γ_f є арифметичною множиною. Звідси маємо: арифметична формула Φ виражає функцію f, якщо Φ виражає предикат "$y=f(x_1, ..., x_n)$".

Приклад 9. Предикати "x є парним числом" та "x ділиться на y". арифметичні, вони виражаються формулами $\exists y(x=y+y)$ та $\exists z(x=y\times z)$.

Приклад 10. Предикат "x є простим числом" арифметичний. Він виражається арифметичною формулою $\forall y\forall z(x=y\times z\rightarrow y=1\vee z=1)\&\neg x=1$.

Приклад 11. Предикати "$x\leq y$" та "$x<y$" арифметичні, бо вони виражаються арифметичними формулами $\exists z(x+z=y)$ та $\exists z(x+z=y\&x\neq y)$.

Використовуючи приклад 4.2.11, в записах арифметичних формул надалі вживатимемо скорочення вигляду $x\leq y$ та $x<y$.

Приклад 12. Предикат "$x\leq y$" в АС $\boldsymbol{N} = (N, \sigma_{ar})$, $\boldsymbol{R} = (R, \sigma_{ar})$ та $\boldsymbol{Z} = (Z, \sigma_{ar})$ виражається різними арифметичними формулами. Справді, для $\boldsymbol{N}$ маємо $\exists z(x+z=y)$; для $\boldsymbol{R}$ маємо $\exists z(x+z\times z=y)$, для $\boldsymbol{Z}$ маємо $\exists z\exists u\exists v\exists w(x+z\times z+u\times u+v\times v+w\times w=y)$.

Приклад 13. Арифметичними є такі функції:

1) Функції $x+y$, $x\times y$ та $x-y$ виражаються відповідно арифметичними формулами $z=x+y$, $z=x\times y$ та $y+z=x$.

2) Функція $[x/y]$ виражається арифметичною формулою

$z\times y \leq x$ & $x<(z+1)\times y$.

3) Функція $\mathrm{mod}(x, y)$ виражається арифметичною формулою

$\exists u(x=z+u\times y$ & $z<y)$.

4) Функція $[\sqrt{x}]$ виражається арифметичною формулою

$z\times z \leq x$ & $x<(z+1)\times(z+1)$.

Приклад 14. Клас арифметичних множин замкнений відносно операцій $\cup$, $\cap$ та доповнення.

Нехай множини A та B виражаються арифметичними формулами Φ та Ψ. Тоді $A\cup B$, $A\cap B$ та $\overline{A}$ виражаються відповідно арифметичними формулами $\Phi\vee\Psi$, $\Phi\&\Psi$ та $\neg\Phi$.

ВПРАВИ

1. Вкажіть формули відповідної мови, що виражають предикати:

1) "$\mathrm{mod}(x,3)=0$" та "$x=2$" в АС $(N; \{+, =\})$;

2) "x парне" та "x непарне" в АС $(Z; \{+, =\})$;

3) "$y=x+9$" в АС $(N, \{y=x+3, =\})$;

4) "$|x-y|=6$" в АС $(R; "|x-y|=2, =\})$;

5) "$x=0$" та "$x=1$" в АС $(N; <, =\})$;

6) "x=1" та "x=–1" в АС (Z; {×, =});

7) "x=1" та "x=z×y" в АС (R; {+, "y=x^2", =}).

2. Вкажіть формули L_{ar}, що виражають предикати:

1) "існує більше чотирьох парних чисел";

2) "існує не менше чотирьох непарних чисел";

3) "існує не більше двох повних кубів";

4) "існує менше трьох повних квадратів";

5) "не існує простих чисел, кратних 4";

6) "існують прості числа, кратні 5";

7) "існує рівно 2 непростих чисел, кратних 4";

8) "існує менше чотирьох повних квадратів, кратних 5";

9) "існує принаймні 3 непарних простих чисел";

10) "існує рівно 3 парних чисел, що є точними кубами";

11) "існує більше двох простих чисел, не кратних 3";

12) "невірно, що існує рівно 2 чисел, що є сумою 4-х квадратів";

13) "існує менше чотирьох парних непростих чисел";

14) "існує більше двох чисел, що є сумою трьох кубів";

15) "множина непарних чисел нескінченна";

16) "існує єдине парне просте число";

17) "кожне парне число, більше за 2, є сумою двох простих чисел".

3. Вкажіть формули L_{ar}, що виражають функції:

1) $x \dot{-} y$;

2) $|x-y|$;

3) $[\sqrt[x]{y}]$;

4) $\mathrm{mod}(x, [y/z])$;

5) $HCD(x, y)$;

6) $HCK(x, y)$;

7) $HCD(x, [y/z])$;

8) $[x / HCK(y, z)]$);

9) $[\sqrt{HCK(x, y)}]$;

10) $\mathrm{mod}(x, [\sqrt{y}])$;

11) $HCD([\sqrt{x}], y)$;

12) $HCK(x, [\sqrt{[y/z]}])$;

13) $[HCD(x, y) / [\sqrt{z}]]$;

14) $[\sqrt{HCD([x/y], z)}]$.

Гомоморфізми алгебраїчних систем. Метод автоморфізмів

Для класичних алгебраїчних систем однієї сигнатури введемо поняття гомоморфізму.

Нехай $\boldsymbol{A} = (A, \boldsymbol{I}, \sigma)$ та $\boldsymbol{B} = (B, \boldsymbol{I}, \sigma)$.

Гомоморфізмом АС $\boldsymbol{A}$ в АС $\boldsymbol{B}$ назвемо відображення $\varphi : A \to B$ таке, що:

– для кожного $f \in Fs$ арності n для всіх $a_1, \ldots, a_n \in A$ маємо

$\varphi(f_A(a_1, \ldots, a_n)) = f_B(\varphi(a_1), \ldots, \varphi(a_n))$ (*HF*)

– для кожного $p \in Ps$ арності n для всіх $a_1, \ldots, a_n \in A$ маємо

якщо $p_A(a_1, \ldots, a_n) = T$, то $p_B(\varphi(a_1), \ldots, \varphi(a_n)) = T$ (*HP*)

Гомоморфізм φ АС $\boldsymbol{A}$ в АС $\boldsymbol{B}$ назвемо *повним гомоморфізмом*, якщо (*HP*) замінюється сильнішою умовою:

– для кожного $p \in Ps$ арності n для всіх $a_1, \ldots, a_n \in A$ маємо

$p_A(a_1, \ldots, a_n) = p_B(\varphi(a_1), \ldots, \varphi(a_n))$

(*EP*)

Повний гомоморфізм φ назвемо *сильним гомоморфізмом*, якщо відображення φ сюр'єктивне.

Повний гомоморфізм φ назвемо *ізоморфізмом*, якщо φ бієктивне.

Якщо сигнатура σ містить символ рівності =, то сильний гомоморфізм стає ізоморфізмом. Справді, тоді (*EP*) гарантує ін'єктивність відображення φ, що за умови сюр'єктивності дає бієктивність φ.

Скажемо, що АС $\boldsymbol{A}$ та АС $\boldsymbol{B}$ *ізоморфні*, що будемо позначати $\boldsymbol{A} \cong \boldsymbol{B}$, якщо існує ізоморфізм АС $\boldsymbol{A}$ в АС $\boldsymbol{B}$.

Таке визначення коректне, бо відношення $\cong$ на множині АС одної сигнатури є відношенням еквівалентності.

Ізоморфізм АС $\boldsymbol{A}$ в АС $\boldsymbol{A}$ назвемо *автоморфізмом* АС $\boldsymbol{A}$.

Приклад 1. Нехай $\boldsymbol{A} = (\{5x \mid x \in N\}, \sigma)$, $\boldsymbol{B} = (\{2x \mid x \in N\}\}, \sigma)$, де $\sigma = \{+, =\}$. Тоді $\varphi(x) = {}^{5}\!/_{2}\, x$ є бієкцією $B \to A$, причому для + та = виконуються (*HF*) та (*EP*). Отже, φ є ізоморфізмом $\boldsymbol{A}$ в $\boldsymbol{B}$.

Відображення $\varphi(x) = {}^{2}\!/_{5}\, x$ є бієкцією $A \to B$, причому для + та = виконуються (*HF*) та (*EP*). Отже, φ є ізоморфізмом $\boldsymbol{B}$ в $\boldsymbol{A}$.

Прикла 2. Задамо $\varphi : N \to \{0, 1\}$ так: $\varphi(x) = \begin{cases} 0, & x \text{ парне}, \\ 1, & x \text{ непарне}. \end{cases}$. Тоді φ – гомоморфізм АС (N, {+, =}) в АС ({0, 1}, {+, =}), бо для + та = виконуються (*HF*) та (*HP*).

Приклад 3. Нехай $\boldsymbol{A} = (Z, \{+, =\})$. Тоді $\varphi(x) = -x$ є бієкцією $Z \to Z$, причому для + та = виконуються умови (*HF*) та (*EP*). Отже, φ є автоморфізмом АС $\boldsymbol{A} = (Z, \{+, =\})$.

Відношення еквівалентності $\approx$ на множині A називають *відношенням конгруентності* на АС $\boldsymbol{A} = (A, \sigma)$, якщо $\approx$ стабільне відносно базових функцій АС $\boldsymbol{A}$. Це означає: для кожного $f \in Fs$ арності n якщо $a_1 \approx b_1, \ldots, a_n \approx b_n$, то $f_A(a_1, \ldots, a_n) \approx f_A(b_1, \ldots, b_n)$.

Нехай $\boldsymbol{A} = (A, \sigma)$, $\approx$ – відношення конгруентності на $\boldsymbol{A}$. *Фактор-система* $\boldsymbol{B} = (B, \sigma)$ системи $\boldsymbol{A}$ за відношенням $\approx$ задається так.

Для кожного $a \in A$ позначимо $[a] = \{c \in A \mid a \approx c\}$. Задамо множину $B = \{[a] \mid a \in A\}$. Таку множину B позначатимемо також $A/\approx$.

Для кожного $f \in Fs$ арності n задамо $f_B([a_1],..., [a_n]) = [f_A(a_1 ,..., a_n)]$.

Для кожного $p \in Ps$ арності n визначимо: $p_B([a_1], ..., [a_n]) = T \Leftrightarrow$ знайдуться $c_1 \in [a_1], ..., c_n \in [a_n]$ такі, що $p_A(c_1, ..., c_n)=T$.

Таке визначення функцій f_B коректне, бо відношення $\approx$ конгруентне.

Справді, для довільних $c_1, ..., c_n$ таких, що $[a_1]=[c_1], ..., [a_n]=[c_n]$, маємо $a_1 \approx c_1, ..., a_n \approx c_n$, звідки $f_A(a_1, ..., a_n) \approx f_A(c_1, ..., c_n)$ за конгруентністю $\approx$, тому $[f_A(a_1, ..., a_n)] = [f_A(c_1, ..., c_n)]$.

Задамо канонічне відображення $\varphi : A \to A/\approx$ таким чином: для кожного $a \in A$ покладемо $\varphi(a)=[a]$.

Канонічне відображення $\varphi : A \to A/\approx$ – гомоморфізм АС $\boldsymbol{A} = (A, \sigma)$ в АС $\boldsymbol{B} = (B, \sigma)$. Такий гомоморфізм називається *канонічним*.

Приклад 4. Нехай $\boldsymbol{N} = (N, \sigma_{ar})$. Задамо відношення $\approx$ на N таким чином: $m \approx n \Leftrightarrow mod(m, 2) = mod(n, 2)$. Таке $\approx$ є відношенням конгруентності на $\boldsymbol{N}$.

Покладемо $N/\approx = \{[0], [1]\}$. Тут [0] та [1] є відповідно множинами парних та непарних натуральних чисел.

ФС + та $\times$ інтерпретуються на $(N/\approx, \sigma_{ar})$ таким чином :

$+([0], [0])=[0]$;	$\times([0], [0])=[0]$;
$+([0], [1])=[1]$;	$\times([0], [1])=[0]$;
$+([1], [0])=[1]$;	$\times([1], [0])=[0]$;
$+([1], [1])=[0]$;	$\times([1], [1])=[1]$.

Константні символи 0 та 1 інтерпретуються як [0] та [1].

Таку фактор-систему $(N/\approx, \sigma_{ar})$ позначимо $\boldsymbol{N}_{mod2}$.

Задамо канонічне відображення $\varphi : N \to N/\approx : \varphi(n) = \begin{cases} [0], & n \text{ парне}, \\ [1], & n \text{ непарне}. \end{cases}$.

Надалі у визначеннях імена & та $\forall x$ похідних логічних операції **&** та $\forall \boldsymbol{x}$ трактуємо як символи розширеного алфавіту. Точне визначення формули мови розширеного алфавіту цілком зрозуміле.

Формулу, утворену з атомарних формул за допомогою символів логічних операцій $\vee$, & та $\exists x$, назвемо $\exists$-*позитивною формулою*.

Формулу, утворену з атомарних формул за допомогою символів логічних операцій $\vee$, &, $\exists x$ та $\forall x$, назвемо *позитивною формулою*.

Кожне $\varphi : A \to B$ можна розширити до відображення $\varphi : A^X \to B^X$:

$\varphi([x_i \mapsto a_i]_{i \in I}) = [x_i \mapsto \varphi(a_i)]_{i \in I})$.

Зокрема, $\varphi((a_1, ..., a_n)) = (\varphi(a_1), ..., \varphi(a_n))$.

Теорема 1. *Нехай* $\varphi : A \to B$ *– гомоморфізм* АС $\boldsymbol{A} = (A, \sigma)$ *в АС* $\boldsymbol{B} = (B, \sigma)$. *Тоді:* 1) *для кожного терма* t *сигнатури* σ *з множиною предметних імен* X *для довільних* $d \in A^X$ *маємо* $\varphi(t_A(d)) = t_B(\varphi(d))$;

2) *для кожної* $\exists$*-позитивної формули* Φ *сигнатури* σ *з множиною вільних імен* X *для довільних* $d \in A^X$ *якщо* $\Phi_A(d)=T$, *то* $\Phi_B(\varphi(d)) = T$.

Теорема 2. *Нехай сюр'єкція* $\varphi : A \to B$ *– гомоморфізм* АС $\boldsymbol{A} = (A, \sigma)$ *в АС* $\boldsymbol{B} = (B, \sigma)$. *Тоді виконується твердження* 1) *теореми 4.3.1 та*

2) *для кожної позитивної формули* Φ *сигнатури* σ *з множиною вільних імен* X *для довільних* $d \in A^X$ *якщо* $\Phi_A(d)=T$, *то* $\Phi_B(\varphi(d)) = T$.

Наслідок. *Нехай* φ *– сюр'єктивний гомоморфізм* АС $\boldsymbol{A} = (A, \sigma)$ *в АС* $\boldsymbol{B} = (B, \sigma)$. *Тоді для кожної позитивної формули* Φ *сигнатури* σ *якщо* $\boldsymbol{A} \models \Phi$, *то* $\boldsymbol{B} \models \Phi$.

Теорема 3 (про ізоморфізм). *Нехай* $\varphi : A \to B$ *– ізоморфізм* АС $\boldsymbol{A} = (A, \sigma)$ *в АС* $\boldsymbol{B} = (B, \sigma)$. *Тоді виконується твердження* 1) *теореми 4.3.1 та*

2) *для кожної формули* Φ *сигнатури* σ *з множиною вільних імен* X *для довільних* $d \in A^X$ *маємо* $\Phi_A(d) = \Phi_B(\varphi(d))$.

Приклад 5. Умова позитивності в формулюванні теорем 4.3.1 та 4.3.2 істотна. Справді, канонічне відображення $\varphi : N \to N/_{\approx}$, де $N/_{\approx} = \{[0], [1]\}$, є сюр'єктивним гомоморфізмом АС $\boldsymbol{N}$ в АС $\boldsymbol{N}_{mod2}$. Розглянемо формулу $\neg(1+1)=0$, яку позначимо Φ. Тоді маємо $\boldsymbol{N} \models \Phi$, але $\boldsymbol{N}_{mod2} \not\models \Phi$, адже $+([1], [1])=[0]$.

АС $\boldsymbol{A} = (A, \sigma)$ та $\boldsymbol{B} = (B, \sigma)$ *елементарно еквівалентні*, якщо для кожної формули Φ сигнатури σ $\boldsymbol{A} \models \Phi \Leftrightarrow \boldsymbol{B} \models \Phi$.

Те, що АС $\boldsymbol{A}$ та $\boldsymbol{B}$ елементарно еквівалентні, позначимо $\boldsymbol{A} \overline{\approx} \boldsymbol{B}$.

Елементарна еквівалентність алгебраїчних систем означає, що вони мають однакові властивості на рівні логіки 1-го порядку, тобто їх ніяк не можна відрізнити, використовуючи мову 1-го порядку.

Приклад 6. $(R; \{\times, =\}) \overline{\not\approx} (Q; \{\times, =\})$. Справді, нехай Φ – формула $\forall x \exists y (x=y \times y \times y)$. Тоді $(R; \{\times, =\}) \models \Phi$ та $(Q; \{\times, =\}) \not\models \Phi$.

Приклад 7. $(Z; \{+, =\}) \overline{\not\approx} (Q; \{+, =\})$. Справді, нехай Φ – формула $\forall x \exists y (x=y+y)$. Тоді $(Q; \{+, =\}) \models \Phi$ та $(Z; \{+, =\}) \not\models \Phi$.

Приклад 8. $(N, \{\leq, =\}) \overline{\not\approx} (Z, \{\leq, =\})$. Справді, нехай Φ – формула $\exists m \forall x (m \leq x)$. Тоді $(N, \{<, =\}) \models \Phi$ та $(Z, \{<, =\}) \not\models \Phi$.

Із теореми 4.3.3 (теореми про ізоморфізм) випливає:

Теорема 4. *Нехай* φ *– ізоморфізм* АС $\boldsymbol{A} = (A, \sigma)$ *в АС* $\boldsymbol{B} = (B, \sigma)$. *Тоді для кожної формули* Φ *сигнатури* σ *маємо* $\boldsymbol{A} \models \Phi \Leftrightarrow \boldsymbol{B} \models \Phi$.

Теорема 5 *Нехай* φ *– автоморфізм* АС $\boldsymbol{A} = (A, \sigma)$. *Тоді для кожної формули* Φ *сигнатури* σ *з множиною вільних імен* X *для довільних* $d \in A^X$ *маємо* $\Phi_A(d) = \Phi_A(\varphi(d))$.

Із теореми 4.3.4 дістаємо, що з ізоморфізму АС $\boldsymbol{A}$ та $\boldsymbol{B}$ випливає їх елементарна еквівалентність. Зворотне, взагалі кажучи, невірне, бо елементарно еквівалентні АС можуть мати носії різної потужності. Наприклад, в [12] доводиться $(Q, \{<, =\}) \overline{\approx} (R, \{<, =\})$, але ці АС неізоморфні, бо множина Q зліченна, а множина R має потужність континууму.

Таким чином, *якщо* $\boldsymbol{A} \cong \boldsymbol{B}$, *то* $\boldsymbol{A} \overline{\approx} \boldsymbol{B}$; *якщо* $\boldsymbol{A} \overline{\not\approx} \boldsymbol{B}$, *то* $\boldsymbol{A} \not\cong \boldsymbol{B}$.

Для доведення виразності предикату P в АС $\boldsymbol{A}$ достатньо вказати таку формулу Φ, що P – це Φ_A. Проте для доведення невиразності предикатів в АС доводиться використовувати потужніші засоби.

Розглянемо метод доведення невиразності предикатів в АС за допомогою автоморфізмів.

Теорема 6. *Нехай* φ – *автоморфізм АС* $\boldsymbol{A} = (A, \sigma)$. *Якщо предикат* $P : A^X \to \{T, F\}$ *виразний в* $\boldsymbol{A}$, *то* $P(d)=P(\varphi(d))$ *для всіх* $d \in A^X$.

Нехай P виразний в $\boldsymbol{A}$ формулою Φ сигнатури σ, тобто P – це Φ_A. Згідно з теоремою 4.3.5 $P(d)=\Phi_A(d)=\Phi_A(\varphi(d))=P(\varphi(d))$ для кожного $d \in A^X$.

Отже, для доведення невиразності P в $\boldsymbol{A}$ досить знайти автоморфізм φ системи $\boldsymbol{A}$ такий, що порушується умова $P(d)=P(\varphi(d))$.

Приклад 9. Предикат "$z=x+y$" не виразний в АС $\boldsymbol{Z}_< = (Z, \{<, =\})$.

Відображення $\varphi(x)=x+1$ є автоморфізмом $\boldsymbol{Z}_<$, бо бієктивне і зберігає значення предикатів < та =. Але $\varphi(0)=1$, тому вірно $0=0+0$ та невірно $\varphi(0)=\varphi(0)+\varphi(0)$.

Приклад 10. Предикат "$x<y$" не виразний в АС $\boldsymbol{Z}_+ = (Z, \{+, =\})$. Справді, відображення $\varphi(x)=-x$ є автоморфізмом $\boldsymbol{Z}_+$, бо бієктивне і зберігає значення функції + та предикату =. Але $\varphi(0)=0$ та $\varphi(1)=-1$, тому вірно $0<1$ та невірно $\varphi(0)<\varphi(1)$.

Приклад 11. Предикат "$y=x+1$" не виразний в АС $(N, y=x+2, =\})$.

Відображення $\varphi(x)=\begin{cases} x+1, & x \text{ парне}, \\ x-1, & x \text{ непарне}, \end{cases}$ є автоморфізмом такої АС, бо бієктивне і зберігає значення базових предикатів. Предикат "$y=x+1$" позначимо $P(x, y)$. Тоді маємо $P(1, 0)=T$ та $P(\varphi(1), \varphi(0)) = P(0, 1)=F$.

Приклад 12. Предикат "$x=2$" не виразний в АС $(N, \{\times, =\})$.

Відображення, задане умовою $\varphi(2^a \cdot 3^b \cdot c)=2^b \cdot 3^a \cdot c$, – автоморфізм такої АС, бо бієктивне і зберігає значення функції $\times$ та предикату =. Предикат "$x=2$" позначимо $P(x)$. Тоді $P(2)=T$ та $P(\varphi(2))=P(3)=F$.

ВПРАВИ

1. Доведіть: відношення $\cong$ на множині АС одної сигнатури є відношенням еквівалентності.

2. Нехай $\varphi : A \to B$ – гомоморфізм АС $\boldsymbol{A} = (A, \sigma)$ в АС $\boldsymbol{B} = (B, \sigma)$. Нехай множина $D \subseteq B$ утворює підсистему системи $\boldsymbol{B}$. Доведіть, що множина $\varphi^{-1}(D) \subseteq A$ утворює підсистему системи $\boldsymbol{A}$.

3. У якому відношенні щодо ізоморфізму такі АС:

1) $(N; \{+, =\})$, $(Z; \{+, =\})$, $(Q; \{+, =\}$, та $(R; \{+, =\})$?

2) $(N; \{<, =\})$, $(Z; \{<, =\})$, $(Q; \{<, =\})$ та $(R; \{<, =\})$?

3) $(N; \{\leq, =\})$, $(Z; \{\leq, =\})$, $(Q; \{\leq, =\})$ та $(R; \{\leq, =\})$?

4. У якому відношенні щодо елементарної еквівалентності такі АС:

1) $(N; \{\times, =\})$, $(Z; \{\times, =\})$, $(Q; \{\times, =\})$, $(R; \{\times, =\})$ та $(C; \{\times, =\})$?

2) $(N; \{+, =\})$, $(Z; \{+, =\})$ та $(R; \{+, =\})$?

3) $(N; \{<, =\})$, $(Z; \{<, =\})$ та $(Q; \{<, =\})$?

4) $(N; \{\leq, =\})$, $(Z; \{\leq, =\})$ та $(R; \{\leq, =\})$?

5. Опишіть всі автоморфізми таких АС:

1) $(R; \{<, =\})$;

2) $(Z; \{<, =\})$;

3) $(Z; \{+, =\})$;

4) $(N; \{+, =\})$;

5) $(N; \{y=x+1, =\})$;

6) $(N; \{|x-y|=1, =\})$.

6. Встановіть, чи виразні в АС предикати:

1) В АС $(Z; \{+, =\})$ предикати

a) "mod $(x, 3)=0$"; "mod $(x, 2)=1$"; "mod $(x, 5)=2$" ?

b) "mod $(x, 4)=0$"; "mod $(x, 3)=2$"; "mod $(x, 4)=3$"; "$z=x\times y$" ?

2) В АС $(Z; \{\times, =\})$ предикати

a) "$x=3$"; "$x=-2$"; " $z=x+y$" ?

b) "$x=1$"; "$x=-1$"; "$x=5$" ?

c) "$x=0$"; "$x=7$"; "$x=y+6$"; "$z=x+y+1$" ?

7. Встановіть, чи виразні в АС предикати:

1) В АС $(N; \{+, =\})$ предикати "$x=3$"; "mod$(x, 3)=0$", "mod$(x, 3)=2$"; "$z=x\times y$" ?

2) В АС $(N; \{\times, =\})$ предикати "$x=10$"; "mod$(x, 3)=0$", "mod$(x, 2)=1$"; "$z=x+y$" ?

3) В АС $(N; \{<, =\})$ предикати "$x=0$"; "$x=2$"; "$x=3$"; "$z=x+1$" ?

4) В АС $(N; \{y=x+1, =\})$ предикати "$x=0$"; "$x=1$"; "$x=2$" ?

5) В АС $(N; \{y=x+2, =\})$ предикати " $x=1$"; "$x=6$"; "$y=x+6$"; "$y=x+1$" ?

6) В АС $(N; \{y=x+3, =\})$ предикати: "$x=3$", "$y=x+2$"; "$y=x+6$"; "$x=2$" ?

7) В АС $(Z; \{y=x+2\})$ предикати "$x=2$"; "$|x-y|=2$"; "$y=x+4$" ?

8) В АС $(Z; \{x-y=3, =\})$ предикати "$x=4$"; "$x-y=4$"; "$|x-y|=6$" ?

9) В АС $(N; \{y=x+4, =\})$ предикати "$x=3$"; "$y=x+3$"; "$|x-y|=12$" ?

10) В АС $(R; \{|x-y|=1, =\})$ предикати "$x=0$"; "$x-y=1$"; "$y=x+2$" ?

11) В АС $(Z; \{|x-y|=1, =\})$ предикати "$|x-y|=2$"; "$y-x=2$"; "$z=x+y$" ?

12) В АС $(Z; \{|x-y|=2, =\})$ предикати "$x=2$"; "$|x-y|=3$"; "$z=x+y$" ?

13) В АС $(Q; \{|x-y|=2, =\})$ предикати "$x-y=2$"; "$z=x+y$"; "$|x-y|=4$"; "$y=x+2$" ?

14) В АС $(Q; \{|x-y|=3, =\})$ предикати "$x=1$"; "$|x-y|=6$"; "$z=x\times y$" ?

15) В АС $(Z; \{|x-y|=3, =\})$ предикати "$x=3$"; "$x-y=3$"; "$|x-y|=12$" ?

16) В АС $(Z; \{|x-y|=4, =\})$ предикати "$x=4$"; "$x-y=4$"; "$|x-y|=8$" ?

17) В АС $(Z; \{x/y, =\})$ предикати "$x=0$"; "$x=1$"; "$z=x+y$" ?

18) В АС $(N; \{x/y, =\})$ предикати "$x=1$"; "$x=5$"; "x – просте" ?

2.3. ПРАКТИЧНА РОБОТА 6. ЧИСЛЕННЯ В ЛОГІКАХ ДРУГОГО ПОРЯДКУ. СЕКВЕНЦІЙНЕ ЧИСЛЕННЯ ЛОГІК ПЕРШОГО ПОРЯДКУ

ВИКОРИСТАНО МАТЕРІАЛ:

1. ***Зубенко В.В., Шкільняк С.С. Основи математичної логіки: навчальний посібник -. К.: НУБіП України, 2020. - 102 с. (Числення першого порядку - сторінка 31-26. Секвенційне числення - сторінка 62-74).***
2. ***С.С. Шкільняк. Математична логіка. Приклади і задачі: Навчальний посібник. – К.: Видавничо-поліграфічний центр "Київський університет", 2007. – 145 с. (сторінка 113--116).***

Теорії першого порядку

Множина логічних аксіом задається такими схемами аксіом:

Ax1) $\neg\Phi\vee\Phi$ – пропозиційні аксіоми;

Ax2) $\Phi_x[t]\to\exists x\Phi$ – аксіоми підстановки;

Ax3) $x=x$ – аксіоми тотожності;

Ax4) $x_1=y_1\to\ldots\to x_n=y_n\to fx_1\ldots x_n=fy_1\ldots y_n$ та

$x_1=y_1\to\ldots\to x_n=y_n\to px_1\ldots x_n\to py_1\ldots y_n$ – аксіоми рівності.

Множина ПВ *P* складається з таких правил виведення:

П1) $\Phi\ |\!-\Psi\vee\Phi$ – правило розширення.

П2) $\Phi\vee\Phi\ |\!-\Phi$ – правило скорочення.

П3) $\Phi\vee(\Psi\vee\Xi)\ |\!-(\Phi\vee\Psi)\vee\Xi$ – правило асоціативності.

П4) $\Phi\vee\Psi,\ \neg\Phi\vee\Xi\ |\!-\Psi\vee\Xi$ – правило перетину.

П5) $\Phi\to\Psi\ |\!-\exists x\Phi\to\Psi$, якщо x не вільна в Ψ, – правило $\exists$-введення.

Логічні аксіоми присутні у всіх теоріях 1-го порядку, власні аксіоми визначають специфіку тієї чи іншої теорії.

Теоремою теорії 1-го порядку *T* називають формулу, яка виводиться із аксіом за допомогою скінченної кількості застосувань ПВ.

Множину теорем теорії *T* позначатимемо Th(*T*). Те, що A – теорема, позначаємо $T\ |\!-A$, або $|\!-A$, якщо *T* мається на увазі.

Абстрагуючись від наборів символів логічних операцій, способів запису термів та формул, наборів логічних аксіом і правил виведення, можна стверджувати, що теорія 1-го порядку визначається сигнатурою мови та множиною власних аксіом.

Сигнатурою теорії 1-го порядку називають сигнатуру мови цієї теорії.

Формулу мови теорії називатимемо також формулою теорії.

Теорія *T'* називається *розширенням* теорії *T*, якщо кожна формула теорії *T* є формулою теорії *T'* та $\mathrm{Th}(T)\subseteq\mathrm{Th}(T')$.

В цьому випадку теорію *T* називають *звуженням* теорії *T'*.

Розширення (звуження) *T'* теорії *T* називають *простим,* якщо *T* та *T'* мають однакові мови.

Теорії 1-го порядку T_1 та T_2 називаються *еквівалентними*, якщо в них однакові мови та множини теорем.

Потужністю теорії T називають потужність множини Th(T). Зокрема, теорія 1-го порядку із зліченною сигнатурою зліченна, теорія 1-го порядку з сигнатурою потужності α має потужність α.

Розглянемо кілька прикладів теорій 1-го порядку.

Приклад 1. Теорія 1-го порядку, яка не містить власних аксіом, називається *численням предикатів 1-го порядку* (скорочено ЧП-1).

Приклад 2. Особливе місце серед формальних теорій займає теорія натуральних чисел – *формальна арифметика*. Позначимо її *Ar.* Мовою *Ar* є мова L_{ar}. Власні аксіоми *Ar* такі:

Ar1) $\neg(x+1=0)$;

Ar2) $x+1=y+1\rightarrow x=y$;

Ar3) $x+0=x$;

Ar4) $x+(y+1)=(x+y)+1$;

Ar5) $x\times 0=0$;

Ar6) $x\times(y+1)=x\times y+x$;

Ar7) $A_x[0]\ \&\ \forall x(A\rightarrow A_x[x+1])\rightarrow \forall xA$ – аксіоми індукції.

Кожна власна аксіома формальної арифметики є ІАФ.

Приклад 3. *Елементарною теорією груп* називається теорія 1-го порядку *Gr* сигнатури {•, e, =}, де e – константний символ, • – бінарний функціональний символ. Власні аксіоми *Gr* такі:

G1) $x\bullet(y\bullet z)=(x\bullet y)\bullet z$;

G2) $\forall x(e\bullet x=x)$;

G3) $\forall x\exists y(y\bullet x=e)$.

Приклад 4. *Елементарною теорією полів* називається теорія 1-го порядку *Fl* сигнатури {0, 1, +, ×, =}, де 0 та 1 – константні символи, + та × – бінарні ФС. Власні аксіоми *Fl* такі:

Fl1) $x+y=y+x$;

Fl2) $x+(y+z)=(x+y)+z$;

Fl3) $x\times y=y\times x$;

Fl4) $x\times(y\times z)=(x\times y)\times z$;

Fl5) $x\times(y+z)=x\times y+x\times z$;

Fl6) $\forall x(x+0=x)$;

Fl7) $\forall x(x\times 1=x)$;

Fl8) $\forall x\exists y(x+y=0)$;

Fl9) $\forall x(\neg(x=0)\rightarrow\exists y(x\times y=1))$;

Fl10) $\neg(0=1)$.

Теорема 1. 1) *Логічні аксіоми є всюди істинними формулами*.

2) *Висновки правил П1–П4 – тавтологічні наслідки засновків*.

3) *Висновок правила П5 – слабкий логічний наслідок засновку*.

Наслідок. *Кожна теорема ЧП-1 є всюди істинною формулою.*

Справді, логічні аксіоми всюди істинні, правила виведення зберігають властивість всюди істинності.

Приклад 5. Умова "x не вільне в Ψ" істотна для П5.

Справді, $x=0\to x=0 \Vdash\neq \exists x(x=0)\to x=0$, бо $x=0\to x=0$ всюди істинна, а в силу $(\exists x(x=0)\to x=0)_N(1)=F$ маємо $\boldsymbol{N} \models\neq \exists x(x=0)\to x=0$.

Приклад 6. Для П5 $\Vdash=$ не можна посилити до $\models=$.

Справді, маємо $x=0\to 1=0 \Vdash=\exists x(x=0)\to 1=0$, але $((x=0\to 1=0)\to(\exists x(x=0)\to 1=0))_N(1)=F$, тому $x=0\to 1=0 \models\neq \exists x(x=0)\to 1=0$.

Моделлю теорії 1-го порядку T називається інтерпретація мови теорії, на якій істинні всі власні аксіоми теорії T.

Приклад 7. Моделлю ЧП-1 є кожна інтерпретація його мови.

Приклад 8. Моделлю елементарної теорії груп Gr є кожна група.

Приклад 9. Моделлю формальної арифметики Ar ***є N*** – стандартна інтерпретація L_{ar}. Таку модель називають *стандартною моделлю* формальної арифметики.

Приклад 10. Моделлю елементарної теорії полів Fl є кожне поле.

Формула Φ називається *істинною* в теорії T, якщо Φ істинна на кожній моделі теорії T.

Теорема 2 (теорема істинності). *Кожна теорема теорії 1-го порядку T істинна в T.*

Теорема 3 (теорема тавтології). *Кожна тавтологія є теоремою.*

Наслідок. *Якщо* $\{\Phi_1, ..., \Phi_n\} \models \Phi$ *та* $\vdash\Phi_1, ..., \vdash\Phi_n$, *то* $\vdash\Phi$.

Приклади виведень в теоріях першого порядку

У наведених прикладах використовуємо теорему тавтології (ТТ).

Приклад 1. $\vdash\forall x A\to A$.

Маємо $\vdash\neg A\to\exists x\neg A$ (аксіома Ах3), звідси за ТТ $\vdash\neg\exists x\neg A\to A$, тобто $\vdash\forall x A\to A$.

Приклад 2 (правило $\forall$-введення). Якщо $\vdash A\to B$ та x не вільне в A, то $\vdash A\to\forall x B$.

Якщо $\vdash A\to B$, то $\vdash\neg B\to\neg A$ за ТТ, звідки $\vdash\exists x\neg B\to\neg A$ за П5. Тоді $\vdash\neg\neg A\to\neg\exists x\neg B$ за ТТ, отже, $\vdash A\to\forall x B$.

Приклад 3 (правило дистрибутивності). Якщо $\vdash A\to B$, то $\vdash\exists x A\to\exists x B$ та $\vdash\forall x A\to\forall x B$.

Маємо $\vdash A\to B$ (умова) та $\vdash B\to\exists x B$ (аксіома Ах3), звідки за ТТ $\vdash A\to\exists x B$, тому за П5 дістаємо $\vdash\exists x A\to\exists x B$.

З умови маємо $\vdash\neg B\to\neg A$ за ТТ, маємо $\vdash\neg A\to\exists x\neg A$ (аксіома Ах3), звідси за ТТ $\vdash\neg B\to\exists x\neg A$. За П5 маємо $\vdash\exists x\neg B\to\exists x\neg A$, тому за ТТ $\vdash\neg\exists x\neg A\to\neg\exists x\neg B$, тобто $\vdash\forall x A\to\forall x B$.

Приклад 4 (правило узагальнення). Якщо $\vdash A$, то $\vdash\forall x A$.

Якщо $\vdash A$, то за П1 $\vdash\forall x A\vee A$, звідки за ТТ $\vdash\neg A\to\forall x A$. Тоді $\vdash\exists x\neg A\to\forall x A$ за П5, тобто $\vdash\forall x A\vee\forall x A$. Тепер $\vdash\forall x A$ за П2.

Приклад 5 (правило уособлення). Якщо $\vdash\forall x A$, то $\vdash A$.

За теоремою 6.2.1 $\vdash\forall x A\to A$. Звідси і з умови $\vdash\forall x A$ за МР $\vdash A$.

Приклад 6 (теорема замикання). $\vdash A \Leftrightarrow \vdash A$.

Випливає з правил узагальнення та уособлення.

Приклад 7 (правило підстановки). Якщо $\vdash$A, то $\vdash \mathrm{A}_{x_1,\ldots,x_n}[t_1,\ldots,t_n]$.

Маємо $\vdash\neg\mathrm{A}_x[t]\to\exists x\neg\mathrm{A}$ (аксіома Ах2), тому $\vdash\forall x\mathrm{A}\to\mathrm{A}_x[t]$ за ТТ. Із $\vdash$A за правилом узагальнення $\vdash\forall x\mathrm{A}$, тому за МР $\vdash\mathrm{A}_x[t]$.

Нехай предметні імена $y_1, \ldots, y_n$ не входять до складу формул A та $\mathrm{A}_{x_1,\ldots,x_n}[t_1, \ldots, t_n]$. Позначимо B формулу $\mathrm{A}_{x_1,\ldots,x_n}[y_1, \ldots, y_n]$.

Тоді згідно з $\vdash\mathrm{A}_x[t]$ послідовно маємо $\vdash \mathrm{A}_{x_1}[y_1]$, $\vdash \mathrm{A}_{x_1,x_2}[y_1, y_2]$, ..., $\vdash \mathrm{A}_{x_1,\ldots,x_n}[y_1, \ldots, y_n]$, $\vdash\vdash \mathrm{B}_{y_1}[t_1]$, ..., $\vdash \mathrm{B}_{y_1,\ldots,y_n}[t_1, \ldots, t_n]$. Але ж формула $\vdash \mathrm{B}_{y_1,\ldots,y_n}[t_1, \ldots, t_n]$ – це і є формула $\mathrm{A}_{x_1,\ldots,x_n}[t_1, \ldots, t_n]$.

На основі прикладів 6.2.2, 6.2.3, 6.2.7 вводимо похідні правила виведення:

правило ∀-вв: $\mathrm{A}\to\mathrm{B} \vdash \mathrm{A}\to\forall x\mathrm{B}$, якщо x не вільне в A;

правило ∀-дис: $\mathrm{A}\to\mathrm{B} \vdash \exists x\mathrm{A}\to\exists x\mathrm{B}$;

правило ∀-дис: $\mathrm{A}\to\mathrm{B} \vdash \forall x\mathrm{A}\to\forall x\mathrm{B}$;

правило підстановки (ПП): $\mathrm{A} \vdash \mathrm{A}_{x_1,\ldots,x_n}[t_1,\ldots,t_n]$.

Приклад 8 (теорема підстановки). $\vdash \mathrm{A}_{x_1,\ldots,x_n}[t_1,\ldots,t_n]\to\exists x_1\ldots\exists x_n\mathrm{A}$ та $\vdash\forall x_1\ldots\forall x_n\mathrm{A}\to \mathrm{A}_{x_1,\ldots,x_n}[t_1,\ldots,t_n]$.

Як Ах3 маємо $\vdash\mathrm{A}\to\exists x_n\mathrm{A}$, ... , $\vdash\exists x_{i+1}\ldots\exists x_n\mathrm{A}\to\exists x_i\,\exists x_{i+1}\ldots\exists x_n\mathrm{A}$, ..., $\vdash\exists x_2\ldots\exists x_n\mathrm{A}\to\exists x_1\ldots\exists x_n\mathrm{A}$, звідки $\vdash\mathrm{A}\to\exists x_1\ldots\exists x_n\mathrm{A}$ за ТТ. Тепер за ПП дістаємо $\vdash \mathrm{A}_{x_1,\ldots,x_n}[t_1,\ldots,t_n]\to\exists x_1\ldots\exists x_n\mathrm{A}$

Використовуючи приклад 6.2.1 замість аксіоми Ах3, аналогічно попередньому маємо $\vdash\forall x_1\ldots\forall x_n\mathrm{A}\to\mathrm{A}$, звідки $\vdash\forall x_1\ldots\forall x_n\mathrm{A}\to \mathrm{A}_{x_1,\ldots,x_n}[t_1,\ldots,t_n]$ за ПП.

Приклад 9 (правило симетрії). Для довільних термів a та b маємо $\vdash a=b\leftrightarrow b=a$.

Маємо $\vdash x=y\to x=x\to x=x\to y=x$ (аксіома рівності для ПС =), звідки $\vdash x=x\to x=x\to y=x\to y=x$ за ТТ. Але $\vdash x=x$ (аксіома тотожності), тому послідовно $\vdash x=x\to y=x\to y=x$ та $\vdash x=y\to y=x$ за МР. Аналогічно $\vdash y=x\to x=y$, тому $\vdash x=y\leftrightarrow y=x$ за ТТ. Звідси за ПП $\vdash a=b\leftrightarrow b=a$.

Приклад 10 (правило транзитивності). Для довільних термів a, b та c маємо $\vdash a=b\to b=c\to a=c$.

Маємо $\vdash y=x\to y=z\to y=y\to x=z$ (аксіома рівності для ПС =), звідки $\vdash y=y\to y=x\to y=z\to x=z$ за ТТ. Але $\vdash y=y$ (аксіома тотожності), тому за МР $\vdash y=x\to y=z\to x=z$. Згідно з правилом симетрії маємо $\vdash x=y\to y=x$, тому $\vdash x=y\to y=z\to x=z$ за ТТ. За ПП дістаємо $\vdash a=b\to b=c\to a=c$.

Нехай Γ – деяка множина формул, T – теорія 1-го порядку. Теорію, утворену із T додаванням формул множини Γ як нових власних аксіом, позначаємо $T[\Gamma]$. Якщо $\Gamma=\{\Psi_1,\ldots,\Psi_n\}$, то замість $T[\{\Psi_1,\ldots,\Psi_n\}]$ пишемо $T[\Psi_1,\ldots,\Psi_n]$. При $\Gamma=\{\Psi\}$ пишемо $T[\Psi]\vdash\Phi$.

Теорема 2 (теорема дедукції). *Нехай* A – *замкнена формула. Тоді для довільної формули* B *маємо*: $T \vdash\mathrm{A}\to\mathrm{B} \Leftrightarrow T[\mathrm{A}] \vdash\mathrm{B}$.

Наслідок. *Нехай* $\mathrm{A}_1, \ldots, \mathrm{A}_n$ – *замкнені формули*. *Тоді для довільної формули* B *маємо* $T \vdash\mathrm{A}_1\to\ldots\to\mathrm{A}_n\to\mathrm{B} \Leftrightarrow T[\mathrm{A}_1, \ldots, \mathrm{A}_n] \vdash\mathrm{B}$.

Приклад 11. Якщо не вимагати замкненості формули A, теорема дедукції невірна. Справді, *Ar* [*x*=0] |–*x*=0, тому *Ar* [*x*=0] |–*y*=0 за ПП. Але формула *x*=0→*y*=0 не є ІАФ, тому невірно *Ar* |–*x*=0→*y*=0.

Теорема 3 (теорема редукції). *Нехай* Γ *– деяка множина формул. Тоді* T [Γ] |–A ⇔ T |–$B_1 \to \ldots \to B_n \to A$ *для деяких* $B_1, \ldots, B_n$, *де кожна формула* B_k *є замиканням деякої формули із* Γ.

Приклад 12 (теорема про варіанту). Якщо A' – варіанта формули A, то |–A↔A'.

Покажемо $\vdash \exists x B \leftrightarrow \exists y B_x[y]$, якщо y не вільна в B. Тоді наше твердження випливатиме з теореми еквівалентності та ТТ. Маємо $\vdash B_x[y] \to \exists x B$ (аксіома Ах3), звідки $\vdash \exists y B_x[y] \to \exists x B$ за П5. Але $(B_x[y])_y[x]$ співпадає з B, адже y не вільне в B. Тому $\vdash B \to \exists y B_x[y]$ як Ах3, звідки $\vdash \exists x B \to \exists y B_x[y]$ за П5. Звідси та з $\vdash \exists y B_x[y] \to \exists x B$ за ТТ $\vdash \exists x B \leftrightarrow \exists y B_x[y]$.

ВПРАВИ

1. Вкажіть виведення в численні предикатів таких формул:

1) $\vdash \exists x A \& B \leftrightarrow \exists x (A \& B)$, якщо *x* не вільна в B;

2) $\vdash \forall x A \& B \leftrightarrow \forall x (A \& B)$, якщо *x* не вільна в B;

3) $\vdash \forall x \forall y A \leftrightarrow \forall y \forall x A$;

4) $\vdash \forall x A \& \forall x B \leftrightarrow \forall x (A \& B)$;

5) $\vdash \forall x (A \& B) \to \forall x A \& B$;

6) $\vdash (A \to \forall x B) \to (\forall x A \to B)$;

7) $\vdash (\forall x A \to \exists x B) \to \exists x (A \to B)$;

8) $\vdash (\exists x A \to \forall x B) \to \forall x (A \to B)$;

9) $\vdash \exists x (A \to B) \to (\forall x A \to \exists x B)$;

10) $\vdash \forall x (A \to B) \to (\forall x A \to \forall x B)$;

11) $\vdash \forall x (A \to B) \to (\exists x A \to \exists x B)$;

12) $\vdash (\forall x A \to \forall x B) \to \exists x (A \to B)$;

13) $\vdash (\exists x A \to \exists x B) \to \exists x (A \to B)$.

2. Встановіть, чи вірно:

1) $\vdash \exists x A \& B \to \exists x (A \& B)$?

2) $\vdash \exists x (A \& B) \to \exists x A \& B$?

3) $\vdash \exists x A \& \exists x B \to \exists x (A \& B)$?

4) $\vdash \exists x(A\&B)\to\exists xA\&\exists xB$?

5) $\vdash \forall xA\vee\forall xB\to\forall x(A\vee B)$?

6) $\vdash \forall x(A\vee B)\to\forall xA\vee\forall xB$?

7) якщо $\vdash \exists xA\&B$, то $\vdash \exists x(A\&B)$?

3. Доведіть теореми про дедукцію та про редукцію.

4. Доведіть синтаксичні варіанти теорем про пренексні операції:

1) $\vdash \neg\forall xB\leftrightarrow\exists x\neg B$ та $\vdash \neg\exists xB\leftrightarrow\forall x\neg B$;

2) $\vdash \exists xB\vee C\leftrightarrow\exists x(B\vee C)$ та $\vdash \forall xB\vee C\leftrightarrow\forall x(B\vee C)$, якщо x не вільна в C;

3) $\vdash B\vee\exists xC\leftrightarrow\exists x(B\vee C)$ та $\vdash B\vee\forall xC\leftrightarrow\forall x(B\vee C)$, якщо x не вільна в B.

5. Доведіть синтаксичні варіанти теореми еквівалентності:

Нехай Φ' отримана із формули Φ заміною деяких входжень формул $B_1, ..., B_n$ на $P_1, ..., P_n$ відповідно. Якщо $\vdash B_1\leftrightarrow P_1, ..., \vdash B_n\leftrightarrow P_n$, то $\vdash \Phi\leftrightarrow\Phi'$.

6. Доведіть синтаксичні варіанти теорем рівності:

1) Нехай Φ' отримана із формули Φ заміною деяких входжень термів $t_1,..., t_n$ на терми $s_1,..., s_n$ відповідно. Якщо $\vdash t_1=s_1, ..., \vdash t_n=s_n$, то $\vdash \Phi\leftrightarrow\Phi'$.

2) Нехай терм τ' отриманий із терма τ заміною деяких входжень термів $t_1,..., t_n$ на терми $s_1,..., s_n$ відповідно. Якщо $\vdash t_1=s_1, ..., \vdash t_n=s_n$, то $\vdash \tau=\tau'$.

7. Нехай Φ' – пренексна форма формули Φ. Доведіть $\vdash \Phi\leftrightarrow\Phi'$.

Секвенційні числення логік предикатів першого порядку

Розглянемо секвенційні числення класичних логік кванторного рівня – чистих логік предикатів 1-го порядку. Такі числення можна трактувати як окремий випадок секвенційних числень НКЛ кванторного рівня.

Враховуючи, що для класичних логік на кванторному рівні базовими є композиції $\vee$, $\neg$ та $\exists x$, для секвенцій відмічених формул можна ввести базові секвенційні форми $\vdash\neg$, $\dashv\neg$, $\vdash\vee$, $\dashv\vee$ та $\vdash\exists$ і $\dashv\exists$:

$$\vdash\exists\ \frac{\vdash A_x[y], \Sigma}{\vdash \exists xA, \Sigma} \qquad \dashv\exists\ \frac{\dashv A_x[z_1],..., \dashv A_x[z_m], \Sigma, \dashv \exists xA}{\dashv \exists xA, \Sigma}.$$

Для $\vdash\exists$ умова: вільна змінна $y\notin\Sigma\cup\{\exists xA\}$.

При застосуванні $\dashv\exists$ $\{z_1,\ldots, z_m\}$ – множина вільних імен множини доступних формул секвенції $\dashv\exists xA, \Sigma$ та її наступників.

Процедура побудови секвенційного дерева для секвенції Σ аналогічна відповідній процедурі для *QZN*-числень. Вкажемо тільки її особливості.

Список *TN* складається з предметних імен, що не зустрічаються в формулах Σ. При застосуванні $\vdash\exists$ беремо y як перше незадіяне на шляху від кореня до даного листа ім'я списку *TN*. При застосуванні $\dashv\exists$

множина $\{z_1,\ldots, z_m\}$ складається з усіх вільних імен доступних формул листа та його наступників.

При побудові секвенційного дерева можливі такі випадки:

1) Процедура завершена позитивно, маємо замкнене дерево.

2) Процедура завершена негативно або не завершується, тоді маємо незамкнене дерево. В такому дереві існує незамкнений шлях $\wp$, який дає змогу визначити контрмодель.

Для секвенційних числень класичних логік кванторного рівня маємо повні аналоги теорем 8.1.1–8.1.4.

Секвенційні числення класичних логік функціонального рівня розглянуті в [6, 13].

Приклад 1. Виведення формули $(\exists x\mathrm{P}(x)\to\mathrm{Q}(x)) \to\mathrm{P}(x)\to\exists x\mathrm{Q}(x)$.

$$\dashv(\exists x\mathrm{P}(x)\to\mathrm{Q}(x)) \to\mathrm{P}(x)\to\exists x\mathrm{Q}(x).$$

$$\downarrow$$

$$\vdash\exists x\mathrm{P}(x)\to\mathrm{Q}(x),\ \dashv\mathrm{P}(x)\to\exists x\mathrm{Q}(x)$$

$$\downarrow \qquad \dashv\exists x\mathrm{P}(x),\ \dashv\mathrm{P}(x)\to\exists x\mathrm{Q}(x) \qquad \downarrow \qquad \vdash\mathrm{Q}(x),\ \dashv\mathrm{P}(x)\to\exists x\mathrm{Q}(x)$$

$$\downarrow \qquad\qquad \downarrow$$

$$\dashv\exists x\mathrm{P}(x),\ \vdash\mathrm{P}(x),\ \dashv\exists x\mathrm{Q}(x) \qquad \vdash\mathrm{Q}(x),\ \vdash\mathrm{P}(x),\ \dashv\exists x\mathrm{Q}(x)$$

$$\dashv\mathrm{P}(x),\ \vdash\mathrm{P}(x),\ \dashv\exists x\mathrm{Q}(x),\ \dashv\exists x\mathrm{P}(x)\ \times \qquad \vdash\mathrm{Q}(x),\ \vdash\mathrm{P}(x),\ \dashv\mathrm{Q}(x),\ \dashv\exists x\mathrm{Q}(x)\ \times$$

Отримали замкнене секвенційне дерево.

Приклад 2. Для встановлення вірності $\mathrm{P}(x)\to\exists x\mathrm{Q}(x) \models \exists x\mathrm{P}(x)\to\mathrm{Q}(x)$ будуємо виведення секвенції $\vdash\mathrm{P}(x)\to\exists x\mathrm{Q}(x),\ \dashv\exists x\mathrm{P}(x)\to\mathrm{Q}(x)$.

$$\vdash\mathrm{P}(x)\to\exists x\mathrm{Q}(x),\ \dashv\exists x\mathrm{P}(x)\to\mathrm{Q}(x)$$

$$\dashv\mathrm{P}(x),\ \dashv\exists x\mathrm{P}(x)\to\mathrm{Q}(x) \qquad \vdash\exists x\mathrm{Q}(x),\ \dashv\exists x\mathrm{P}(x)\to\mathrm{Q}(x)$$

$$\dashv\mathrm{P}(x),\ \vdash\exists x\mathrm{P}(x),\ \dashv\mathrm{Q}(x) \qquad \vdash\exists x\mathrm{Q}(x),\ \vdash\exists x\mathrm{P}(x),\ \dashv\mathrm{Q}(x)$$

$$\dashv\mathrm{P}(x),\ \vdash\mathrm{P}(u),\ \dashv\mathrm{Q}(x) \qquad \vdash\mathrm{Q}(u),\ \vdash\exists x\mathrm{P}(x),\ \dashv\mathrm{Q}(x)$$

$$\vdash\mathrm{Q}(u),\ \vdash\mathrm{P}(v),\ \dashv\mathrm{Q}(x)$$

Отримали незамкнене секвенційне дерево, тому невірно, що $\mathrm{P}(x)\to\exists x\mathrm{Q}(x)\models\exists x\mathrm{P}(x)\to\mathrm{Q}(x)$.

Для лівого незамкненого шляху отримуємо контрмодель $\boldsymbol{A}$, для якої $\delta = [x\mapsto a,\ u\mapsto b]$, $\mathrm{P}_A(a)=F$, $\mathrm{P}_A(b)=T$, $\mathrm{Q}_A(a)=F$.

Для правого незамкненого шляху маємо контрмодель $\boldsymbol{B}$, для якої $\delta = [x\mapsto a,\ u\mapsto b,\ v\mapsto c]$, $\mathrm{Q}_B(b)=T$, $\mathrm{P}_B(c)=T$, $\mathrm{Q}_B(a)=F$.

Приклад 3. Для встановлення вірності $\exists x\forall y\mathrm{P}(x,y)\models\forall y\exists x\mathrm{P}(x,y)$ будуємо виведення секвенції $\vdash\exists x\forall y\mathrm{P}(x,y)$, $\dashv\forall y\exists x\mathrm{P}(x,y)$.

$\vdash \exists x \forall y P(x,y), \dashv \forall y \exists x P(x,y)$

$\downarrow$

$\vdash \forall y P(u,y), \dashv \forall y \exists x P(x,y)$

$\downarrow$

$\vdash \forall y P(u,y), \dashv \exists x P(x,v)$

$\downarrow$

$\vdash P(u,u), \vdash P(u,v), \vdash \forall y P(u,y), \dashv \exists x P(x,v)$

$\vdash P(u,u), \vdash P(u,v), \vdash \forall y P(u,y), \dashv P(u,v), \dashv P(v,v) \dashv \exists x P(x,v)$ ×

Отримали замкнене секвенційне дерево.

ВПРАВИ

1. Для секвенційних числень класичних логік кванторного рівня вкажіть похідні секвенційні форми $\vdash\forall$ та $\dashv\forall$.

2. Збудуйте в секвенційному численні класичних логік кванторного рівня виведення чи доведіть його відстутність для таких формул:

1) $\forall x(A\to B)\to(\exists xA\to \exists xB)$;

2) $\forall x(A\to B)\to(\forall xA\to\forall xB)$;

3) $\forall x(A\to B)\to(\exists xA\to\forall xB)$;

4) $\exists x(A\to B)\to(\forall xA\to \exists xB)$;

5) $\exists x(A\to B)\to(\forall xA\to\forall xB)$;

6) $\exists x(A\to B)\to(\exists xA\to\exists xB)$;

7) $(\forall xA\to\forall xB)\to\forall x(A\to B)$;

8) $(\forall xA\to\forall xB)\to\exists x(A\to B)$;

9) $(\exists xA\to\exists xB)\to\forall x(A\to B)$;

10) $(\exists xA\to\forall xB)\to\forall x(A\to B)$;

11) $\forall xA\&B\to\forall x(A\&B)$;

12) $\exists x(A\vee B)\to\exists xA\vee B$;

13) $\forall x\exists yA\to\exists y\forall xA$;

14) $\exists x(A\vee B)\leftrightarrow\exists xA\vee\exists xB$;

15) $\forall x(A\&B)\leftrightarrow\forall xA\&\forall xB$;

16) $\exists x(A\&B)\to\exists xA\&\exists xB$;

17) $\exists x(A\vee\exists xB)\to\exists x\neg A$.

3. Доведіть теореми коректності та повноти для секвенційних числень класичних логік кванторного рівня.

2.4.ПРАКТИЧНА РОБОТА 7. ЕКВІВАЛЕНТНІ ПЕРЕТВОРЕННЯ ФОРМУЛ. РЕЗОЛЮЦІЙНЕ ЧИСЛЕННЯ ЛОГІКИ ПЕРШОГО ПОРЯДКУ

ВИКОРИСТАНО МАТЕРІАЛ:

1. ***Зубенко В.В., Шкільняк С.С. Основи математичної логіки: навчальний посібник -. К.: НУБіП України, 2020. - 102 с. (Побудова пренексної форми – сторінка Резолюційне числення першого порядку – сторінка 31-37.)***
2. ***С.С. Шкільняк. Математична логіка. Приклади і задачі: Навчальний посібник. – К.: Видавничо-поліграфічний центр "Київський університет", 2007. – 145 с. (Побудова пренексної форми – сторінка Резолюційне числення першого порядку - сторінка 83-86)***

Еквівалентні перетворення формул

Основою еквівалентних перетворень формул є семантична теорема еквівалентності.

Теорема 1. *Нехай* A' *отримана із формули* A *заміною деяких входжень формул* $B_1, \ldots, B_n$ *на* $P_1, \ldots, P_n$ *відповідно. Якщо* $B_1 \sim P_1, \ldots, B_n \sim P_n$, *то* $A \sim A'$.

Формула A' називається *варіантою* формули A, якщо A' можна отримати із A послідовними замінами такого типу: підформулу $\exists x B$ замінюємо на $\exists y B_x[y]$, де y не вільна в B.

Теорема 2. (про варіанту). *Якщо* A' – *варіанта формули* A, *то* $A \sim A'$.

Формула A знаходиться в *пренексній формі*, якщо вона має вигляд $Qx_1 \ldots Qx_n B$, де Qx_k – кванторний префікс $\exists x_k$ або $\forall x_k$, B – безкванторна формула, яку називають *матрицею* формули A.

Формулу в пренексній формі називають *пренексною формулою*.

Зауважимо, що коли мовиться про пренексну форму, то $\forall$ не прийнято виражати через $\neg$ та $\exists$.

Теорема 3. 1) $\neg\forall x B \sim \exists x \neg B$ *та* $\neg\exists x B \sim \forall x \neg B$;

2) $\exists x B \vee C \sim \exists x (B \vee C)$ та $\forall x B \vee C \sim \forall x (B \vee C)$, *якщо x не вільна в* C;

3) $B \vee \exists x C \sim \exists x (B \vee C)$ та $B \vee \forall x C \sim \forall x (B \vee C)$, *якщо x не вільна в* B.

Введемо пренексні операції над формулами, які дозволять кожну формулу перетворити до еквівалентної їй пренексної формули. Ці операції грунтуються на теоремах 4.1.3 та 4.1.4.

Під *пренексними операціями* над формулою A розуміємо такі операції:

a) заміна A деякою її варіантою;

b) заміна в A підформул вигляду $\neg\exists x B$ та $\neg\forall x B$ на $\forall x \neg B$ та $\exists x \neg B$ відповідно;

c) заміна в A підформул вигляду $Qx B \vee C$ на $Qx(B \vee C)$, якщо x не вільне в C; заміна в A підформул вигляду $B \vee Qx C$ на $Qx(B \vee C)$, якщо x не вільне в B.

Пренексною формою формули A назвемо пренексну формулу A', утворену із A за допомогою пренексних операцій.

Теорема 4. *Кожна формула має пренексну форму, причому якщо* A' – *пренексна форма формули* A, *то* $A \sim A'$.

Розглянутий метод побудови пренексної форми передбачає роботу в системі логічних операцій $\{\neg, \vee, \exists x, \forall x\}$. Для уникнення елімінації логічних зв'язок & та $\rightarrow$ можна ввести додаткові пренексні операції:

d) заміна в A підформул вигляду $Qx B \& C$ на $Qx(B \& C)$, якщо x не вільне в C, та підформул вигляду $B \& Qx C$ на $Qx(B \& C)$, якщо x не вільне в B;

e) заміна в A підформул вигляду $B \to QxC$ на $Qx(B \to C)$, якщо x не вільне в B;

f) заміна в A підформул вигляду $\exists xB \to C$ на $\forall x(B \to C)$, та підформул вигляду $\forall xB \to C$ на $\exists x(B \to C)$, якщо x не вільне в C.

Виконання кожної з операцій типу d), e) чи f) зводиться до виконання певної послідовності операцій b) та c). У той же час для $\leftrightarrow$ подібних операцій нема.

Приклад 5. Знайдемо пренексну форму для формули $\exists z(x=y+z) \to (x=y) \vee \exists z((x=y+z) \& \neg(z=0))$:

$\exists z(x=y+z) \to (x=y) \vee \exists t((x=y+t) \& \neg(t=0))$ – операція a);

$\exists z(x=y+z) \to \exists t(x=y) \vee (x=y+t) \& \neg(t=0)$ – операція c);

$\forall z((x=y+z) \to \exists t(x=y) \vee (x=y+t) \& \neg(t=0))$ – операція f);

$\forall z \exists t((x=y+z) \to (x=y) \vee (x=y+t) \& \neg(t=0))$ – операція e).

ВПРАВИ

Вкажіть пренексну форму для таких формул:

1) $\forall xA(x) \to \forall y(\exists zB(x,y,z) \to \neg \forall xA(x) \& \exists xC(x,y))$;

2) $\forall x \neg \exists yA(x,y) \to \forall xB(x) \to \neg \exists yA(x,y)$;

3) $\neg \forall xA(x) \& \exists xB(x) \vee \forall x(\forall yC(x,y) \to A(y))$;

4) $\forall xA(x) \to \forall y(\forall zB(x,y,z) \to \neg \forall xA(x))$;

5) $\forall x \forall yA(x,y) \& \exists xB(x) \to \neg \exists xA(x,y)$.

6) $\exists x \neg \forall yA(x, y) \to \neg \exists xB(x, y) \to \exists yA(x, y)$.

7) $\neg \forall xA(x) \vee \neg \exists xB(x) \vee \forall x(\forall yC(x, y, z) \to A(y))$.

8) $\exists xA(x, y) \to \forall y(\forall zB(x, y, z) \to \neg \forall x \forall yA(x, y))$.

9) $\forall x \exists yA(x, y, z) \& \neg \exists xB(x, y) \to \forall xC(x, y)$.

10) $\exists xA(x) \to \exists y(\forall xB(x, y) \vee \neg \forall zC(x, z) \to \neg \exists xC(x, y))$.

11) $\exists z(x=y+z) \to (x=y) \vee \exists z((x=y+z) \& \neg(z=0))$.

Метод резолюцій логіки 1-го порядку

Основою методу резолюцій для логіки предикатів 1-го порядку є теорема Ербрана.

Комбінацію кванторів $\forall x \exists y$ можна трактувати як твердження про існування певної функції, значення y якої залежить від x. Алгоритмічним уточненням такого принципу для класичної логіки предикатів можна вважати теорему Ербрана

Нехай $\boldsymbol{Q}^{\bar{\boldsymbol{v}}} M(\bar{\boldsymbol{v}})$ – замкнена пренексна формула. Тут $\boldsymbol{Q}^{\bar{\boldsymbol{v}}}$ – кванторні префікси (всі ці кванторні префікси – за різними предметними іменами), $\bar{\boldsymbol{v}}$ – всі вільні предметні імена (змінні) безкванторної формули M, причому $\bar{\boldsymbol{v}}$ складається з $\exists$-кванторних імен $y_1, \ldots, y_n$ та $\forall$-кванторних імен $x_1, \ldots, x_m$. Зіставимо кожному кванторному префіксу $\exists y_i$ із $\boldsymbol{Q}^{\bar{\boldsymbol{v}}}$ $\{\bar{\boldsymbol{x}}_{\boldsymbol{i}}\}$-арну функцію f_i, де $\bar{\boldsymbol{x}}_{\boldsymbol{i}}$ – всі ті $\forall$-кванторні імена із $\bar{\boldsymbol{v}}$, що передують y_i в $\boldsymbol{Q}^{\bar{\boldsymbol{v}}}$. Функції f_i зіставимо новий функціональний символ f_i, арність якого рівна кількості змінних в $\bar{\boldsymbol{x}}_{\boldsymbol{i}}$. Якщо y_i не передує в $\boldsymbol{Q}^{\bar{\boldsymbol{v}}}$ жодний $\forall$-кванторний префікс, то f_i – константа, f_i – константний символ.

Замінимо всі входження y_i в M на терм $f_i(\bar{x}_i)$, $i \in \{1,\dots n\}$. В результаті отримаємо формулу $\forall x_1 \dots \forall x_m M(x_1, \dots, x_m, f_1(\bar{x}_1), \dots, f_n(\bar{x}_n))$. Таке перетворення називається сколемізацією [4, 6, 13], а самі формули зазначеного вигляду – сколемівськими, або формулами в сколемівській формі.

Приклад 1. Нехай початкова формула має вигляд $\forall x\, p(x,x) \,\&\, \forall x \exists y (q(y) \to p(x,y)) \,\&\, \forall y \exists x (p(x,y))$. Зводячи її до пренексної форми, отримуємо $\forall x \forall z \exists y \forall u \exists v (p(x,x) \,\&\, (q(y) \to p(z,y)) \,\&\, p(v,u))$. Тепер $\exists y$ зіставимо 2-арний функціональний символ f, $\exists v$ зіставимо 3-арний функціональний символ g, замінимо входження y термом $f(x,y)$, входження v – термом $g(x,y,u)$. В результаті отримаємо сколемівську формулу $\forall x \forall z \forall u (p(x,x) \,\&\, (q(f(x,y)) \to p(z, f(x,y))) \,\&\, p(g(x,y,u),u))$.

Приклад 2. Нехай формула має вигляд $\exists x\, p(x) \,\&\, \forall x (p(x) \to \exists y (p(y))$. Зводячи її до пренексної форми, отримуємо $\exists x \forall z \exists y (p(x) \,\&\, (p(z) \to p(y))$. Тепер $\exists x$ зіставимо константний символ c, $\exists y$ зіставимо 1-арний функціональний символ f, замінимо входження x константним символом c, входження y – термом $f(z)$. У результаті отримаємо сколемівську формулу $\forall z (p(c) \,\&\, (p(z) \to p(f(z)))$.

Для формули в сколемівській формі збудуємо універс із замкнених термів, отриманих всіма можливими застосуваннями сколемівських функціональних символів до сколемівських константних символів. Якщо таких константних символів немає, як у прикладі 9.2.1, беремо довільний новий константний символ. Такий універс вперше увів до розгляду французький математик Ж. Ербран (1909–1931), тому його називають ербранівським універсом.

Приклад 3. Для сколемівської функції прикладу 9.2.1 беремо новий константний символ a, тоді ербранівський універс – це множина термів $\{a, f(a,a), g(a,a,a), f(f(a,a),a), f(a,f(a,a)), g(f(a,a),a,a), g(a,f(a,a),a), g(a,a,f(a,a)), f(g(a,a,a),a), f(a,g(a,a,a)), g(g(a,a,a),a,a), g(a,g(a,a,a),a), g(a,a,g(a,a,a)), f(f(a,a),f(a,a)), f(g(a,a,a), f(a,a)), f(f(a,a),g(a,a,a)), f(g(a,a,a)), g(a,a,a)), \dots\}$.

Приклад 4. Для сколемівської функції прикладу 9.2.2 ербранівський універс – це множина термів $\{c, f(c), f(f(c)), f(f(f(c))), f(f(f(f(c)))), \dots\}$.

Прикладом замкненої формули $\forall x_1 \dots \forall x_m \Phi(x_1, \dots, x_m)$, де Φ – безкванторна формула, назвемо формулу $\Phi_{x_1,\dots,x_m}[t_1,\dots,t_m]$, де $t_1,\dots,t_m$ – замкнені терми.

Теорема 1 (Ербран). *Пренексна формула $Q^{\bar{v}} M(\bar{v})$ є суперечністю $\Leftrightarrow$ існують системи термів $t_1^1,\dots, t_m^1, \dots, t_1^k,\dots,t_m^k$ ербранівського універсу сколемівської формули $\forall x_1 \dots \forall x_m \Phi$, де Φ – це формула $M(x_1, \dots, x_m, f_1(\bar{x}_1), \dots, f_n(\bar{x}_n))$, така, що формула $\Phi_{x_1,\dots,x_m}[t_1^1,\dots,t_m^1] \,\&\, \dots \,\&\, \Phi_{x_1,\dots,x_m}[t_1^k,\dots,t_m^k]$ є суперечністю пропозиційної логіки.*

Формулу $\Phi_{x_1,\dots,x_m}[t_1^1,\dots,t_m^1] \,\&\, \dots \,\&\, \Phi_{x_1,\dots,x_m}[t_1^k,\dots,t_m^k]$ – кон'юнкцію прикладів формули Φ – С.Кліні назвав ербранівською розгорткою формули $Q^{\bar{v}} M(\bar{v})$.

Розглянемо дуальне формулювання теореми Ербрана. Для цього зробимо інверсію сколемизації замкненої пренексної формули $Q^{\bar{v}} M(\bar{v})$, де $\bar{v}$ – із $\exists$-кванторних імен $y_1,\dots,y_n$ та $\forall$-кванторних імен $x_1,\dots,x_m$. Зіставимо з кожним $\forall x_i$ із $Q^{\bar{v}}$ новий функціональний символ g_i, арність якого рівна кількості $\exists$-кванторних імен, що передують x_i в $Q^{\bar{v}}$. Нехай $\bar{y}_i$ – такі $\exists$-кванторні імена. Якщо імені x_i не передує в Q

$\bar{\boldsymbol{v}}$ жодний $\exists$-кванторний префікс, то g_i – константний символ. Замінимо всі входження x_i в M на терм $g_i(\bar{\boldsymbol{y}}_{\boldsymbol{i}})$, $i \in \{1,\dots m\}$. У результаті отримаємо формулу вигляду $\exists y_1 \dots \exists y_n M(g_1(\bar{\boldsymbol{y}}_1), \dots, g_m(\bar{\boldsymbol{y}}_{\boldsymbol{m}}), y_1, \dots, y_n)$.

При інверсній сколемізації теж будується ербранівський універс та справджується

Теорема 2 (Ербран, дуальна форма). *Пренексна формула* $\boldsymbol{Q}^{\bar{\boldsymbol{v}}} M(\bar{\boldsymbol{v}})$ *всюди істинна* $\Leftrightarrow$ *існують системи термів* $t_1^1, \dots, t_n^1, \dots, t_1^k, \dots, t_n^k$ *ербранівського універсу сколемівської формули* $\exists y_1 \dots \exists y_n \Phi$, *де* Φ *– це формула* $M(g_1(\bar{\boldsymbol{y}}_1), \dots, g_m(\bar{\boldsymbol{y}}_{\boldsymbol{m}}), y_1, \dots, y_n)$, *такі, що формула* $\Phi_{y_1,\dots,y_n}[t_1^1, \dots, t_n^1] \vee \dots \vee \Phi_{y_1,\dots,y_n}[t_1^k, \dots, t_n^k]$ *– тавтологія*.

На теоремі Ербрана базується *метод спростування* Ербрана.

Маємо $\models \Phi \Leftrightarrow \models \Psi$ (де Ψ – замикання Φ) $\Leftrightarrow \neg\Psi$ суперечність $\Leftrightarrow$ існує ербранівська розгортка, що є суперечністю.

Таким чином, для встановлення $\models \Phi$ будуємо сколемівську форму для пренексної форми формули $\neg\Psi$, далі поступово породжуємо терми ербранівського універсу та перевіряємо ербранівські розгортки на суперечність.

Істотним для методу резолюцій є пошук контрарних пар літер для диз'юнктів. Це нескладно для пропозиційної логіки, але на кванторному рівні такий пошук стає проблематичним.

Приклад 5. Нехай $D_1 = A(x) \vee B(x)$ та $D_2 = \neg A(f(y)) \vee C(y)$ –диз'юнкти. Зрозуміло, що в D_1 та D_2 немає контрарних літер. Але можна зробити підстановку терма $f(y)$ замість x в D_1.

Розглянемо диз'юнкти $D'_1 = A(f(y)) \vee B(f(y))$ та D_2. В цих диз'юнктах літери $A(f(y))$ та $\neg A(f(y))$ вже контрарні.

Отже, із D'_1 та D_2 можна отримати резольвенту $R = B(f(y)) \vee C(y)$.

Зрозуміло, що, підставляючи різні терми замість y в R, отримаємо інші резольвенти, які можна отримати із D_1 та D_2 правилом резолюцій. У цьому плані R – найзагальніша, інші резольвенти є її конкретизаціями. Таким чином, для отримання резольвент необхідно робити певні підстановки термів.

Нехай L – мова 1-го порядку з множиною предметних імен (змінних) V та множиною термів Tr.

Підстановкою назвемо довільне однозначне відображення $\alpha : V \to Tr$.

Інакше кажучи, підстановка – це іменна множина $\alpha \in {}^V Tr$.

Якщо така іменна множина порожня, підстановка *тотожна*.

Якщо така іменна множина скінченна, підстановка *скінченна*.

Надалі будемо опускати в підстановках компоненти вигляду $x \mapsto x$, що відповідають тотожним замінам.

Прикладом терма t за підстановкою α, або α-*прикладом* терма t, назвемо терм $\alpha(t)$, отриманий із терма t паралельними замінами всіх входжень предметних імен $x \in nm(t)$ на терми $\alpha(x)$ при умові $\alpha(x)\downarrow$. Якщо $\alpha(x)\uparrow$, заміни імені x немає. Це можна трактувати як тотожну заміну.

Приклад 6. Нехай $t = g(x, h(x, y, z))$, $\alpha = [x \mapsto a, y \mapsto f(b), y \mapsto c]$. Тоді $\alpha(t) = g(a, h(a, f(b), c))$.

Нехай $\alpha = [x_i \mapsto t_i]_{i \in I}$ та $\beta = [y_j \mapsto s_j]_{j \in J}$ – підстановки. Добутком (композицією) підстановок α та β назвемо підстановку $\alpha * \beta = \beta \nabla [x_i \mapsto \beta(t_i)]_{i \in I}$.

Приклад 7. Нехай $\alpha = [x \mapsto f(y), y \mapsto z]$, $\beta = [x \mapsto a, y \mapsto b, z \mapsto y]$. Тоді маємо $\alpha * \beta = [x \mapsto a, y \mapsto b, z \mapsto y] \nabla [x \mapsto \beta(f(y)), y \mapsto \beta(z)] = [x \mapsto f(b), z \mapsto y]$. Тут опущена компонента $y \mapsto y$.

Підстановка α називається *уніфікатором* для множини термів $\{t_1, \ldots, t_n\}$, якщо $\alpha(t_1) = \alpha(t_2) = \ldots = \alpha(t_n)$. У цьому випадку кажуть, що множина термів $\{t_1, \ldots, t_n\}$ уніфікується підстановкою α.

Множина термів *уніфікується,* якщо для неї існує уніфікатор.

Уніфікатор υ – *найзагальніший уніфікатор* (НЗУ) множини термів $\{t_1, \ldots, t_n\}$, якщо для кожного уніфікатора α цієї множини існує підстановка η така, що $\alpha = \upsilon * \eta$.

Приклад 8. Нехай маємо множину термів $\{x, f(y)\}$. Тоді $\upsilon = [x \mapsto f(y)]$ – найзагальніший уніфікатор цієї множини. Справді, уніфікатори даної множини мають вигляд $\alpha = [x \mapsto f(t), y \mapsto t]$. Але тоді для α підходить підстановка $\eta = [y \mapsto t]$.

Справді, $\alpha = [y \mapsto t] \nabla [x \mapsto f(t)] = [y \mapsto t] \nabla [x \mapsto \eta(f(y))] = [x \mapsto f(y)] * [y \mapsto t]$.

Поняття уніфікатора можна узагальнити на множини пар термів, що дозволяє побудувати алгоритм уніфікації – алгоритм знаходження найзагальнішого уніфікатора. На даний час відомі досить ефективні алгоритми уніфікації. Детальніше про це написано в [4, 22].

Поняття прикладу, підстановки та НЗУ природним чином продовжується на літери (атомарні формули чи їх заперечення) та диз'юнкти – диз'юнкції літер.

Метод резолюцій логіки предикатів 1-го порядку базується на зведенні формул до сколемівської форми та перетворенні матриці до кон'юнктивної нормальної форми (КНФ). Така КНФ є кон'юнкцією диз'юнктів. При цьому використовуються такі семантичні властивості:

– $\models \forall \bar{\boldsymbol{v}} \, (K_1 \& \ldots \& K_n) \Leftrightarrow \models \forall \bar{\boldsymbol{v}} \, K_1 \& \ldots \& \forall \bar{\boldsymbol{v}} \, K_n \Leftrightarrow \models K_1 \& \ldots \& K_n$;

– правило підстановки $\models \Phi \Leftrightarrow \models \Phi_x[t]$ (дає змогу робити уніфікацію).

Основою методу резолюцій є *правило бінарної резолюції.*

Нехай D_1 та D_2 – диз'юнкти, в яких $nm(\mathrm{D}_1) \cap nm(\mathrm{D}_2) = \varnothing$. Нехай L_1 – літера D_1, L_2 – літера D_2, причому L_1 та L'_2, де $L_2 = \neg L'_2$, або L_2 та L'_1, де $L_1 = \neg L'_1$, мають НЗУ υ. Тоді диз'юнкт $(\upsilon(\mathrm{D}_1) \setminus \upsilon(L_1)) \vee (\upsilon(\mathrm{D}_2) \setminus \upsilon(L_2))$ називають *бінарною резолюцією* диз'юнктів D_1 та D_2.

Літери L_1 та L_2 називають *відрізними*.

Тут позначення $\mathrm{D} \setminus \mathrm{D}'$ означає диз'юнкт, що є диз'юнкцією тих літер D, які не входять до D'.

Крім правила бінарної резолюції, використовується *правило склеювання*.

Нехай в диз'юнкті D не менше 2-х літер з однаковими предикатними символами мають НЗУ υ. Диз'юнкт $\upsilon(\mathrm{D})$ називають *склеюванням* диз'юнкта D.

У загальному вигляді *правило резолюцій* має два диз'юнкти-засновки та один диз'юнкт-висновок – резольвенту засновків.

Диз'юнкт D – *резольвента* диз'юнктів D_1 та D_2, якщо D є однією з бінарних резольвент:

1) D_1 та D_2;

2) D_1 та склеювання чи прикладу D_2;

3) склеювання чи прикладу D_1 та D_2;

4) склеювання чи прикладу D_1 та склеювання чи прикладу D_2.

Приклад 9. Нехай маємо диз'юнкт $\mathrm{D} = p(x) \vee p(g(y)) \vee \neg q(x)$. Його літери $p(x)$ та $p(g(y))$ мають НЗУ $\upsilon = [x \mapsto g(y)]$.

Диз'юнкт $\upsilon(\mathrm{D}) = p(g(y)) \vee \neg q(g(y))$ – склеювання диз'юнкта D.

Приклад 10. Нехай задані диз'юнкти $D_1 = \neg p(x, f(y)) \vee r(y)$ та $D_2 = p(g(a), x) \vee q(x)$. Маємо $x \in nm(D_1) \cap nm(D_2)$, тому візьмемо α-приклад $\alpha(D_2) = p(g(a), u) \vee q(u)$, де $\alpha = [x \mapsto u]$. Позаяк $p(x, f(y))$ та $p(g(a), u)$ мають НЗУ $\upsilon = [x \mapsto g(a), u \mapsto f(y)]$, то маємо $\upsilon(D_1) = \neg p(g(a), f(y)) \vee r(y)$, $\upsilon(\alpha(D_2)) = p(g(a), f(y)) \vee q(f(y))$. Бінарною резольвентою диз'юнктів D_1 та $\alpha(D_2)$ буде $D = r(y) \vee q(f(y))$.

Отже, диз'юнкт D – резольвента диз'юнктів-засновків D_1 та D_2.

Приклад 11. Нехай маємо диз'юнкти $D_1 = p(x) \vee p(g(y)) \vee q(f(y))$ та $D_2 = \neg p(g(f(a))) \vee p(b)$. Склеюванням диз'юнкта D_1 по НЗУ $[x \mapsto g(y)]$ буде $D'_1 = p(g(y)) \vee q(f(y))$. Бінарною резольвентою диз'юнктів D'_1 та D_2 по НЗУ $[y \mapsto f(a)]$ буде $D = q(g(f(a))) \vee p(b)$.

Отже, D – резольвента диз'юнктів-засновків D_1 та D_2.

Теоретично обґрунтовує метод резолюцій

Теорема 3 (теорема повноти методу резолюцій). *Множина диз'юнктів* S *суперечлива* $\Leftrightarrow$ *існує резолютивне виведення* 0 *із* S.

ВПРАВИ

1. Зведіть до сколемівської форми із матрицею в КНФ такі формули:

1) $\forall x \neg p(x, x) \,\&\, \forall x \exists y\, p(x, y) \,\&\, \forall x \forall z \forall y (p(x, z) \,\&\, p(z, y) \to p(x, y))$;

2) $\forall x \neg q(x, x) \,\&\, \forall x \forall z \exists y (q(x, y) \to q(x, z) \,\&\, q(z, y)) \,\&\, \forall x \forall y \forall z (q(x, y) \,\&\, q(y, z) \to q(x, z))$;

3) $\exists x (p(x) \,\&\, \forall y (q(y) \to h(x, y)) \,\&\, \forall x (p(x) \to \forall y (r(y) \to \neg h(x, y))) \to \forall x (q(x) \to \neg r(x))$.

2. Доведіть, що $(\alpha * \beta) * \gamma = \alpha * (\beta * \gamma)$.

3. Які з наведених множин мають НЗУ? Знайдіть такі НЗУ, якщо вони є:

1) $\{p(x), p(y)\}$;

2) $\{q(x, y), q(x, x)\}$;

3) $\{q(x, y, f(y)), q(x, z, z)\}$;

4) $\{p(x, y, z), p(u, f(u, v), v)\}$;

5) $\{q(x, y, f(y), z, g(y, z), v), q(a, h(a), b, s(a, b), u, h(a, b, u))\}$.

4. Побудуйте всі можливі резольвенти (якщо вони є) для таких пар

1) $D_1 = p(x, y) \vee \neg q(x)$ та $D_2 = p(z, y) \vee q(z)$;

2) $D_1 = \neg p(x) \vee q(x, x)$ та $D_2 = \neg q(y, f(y))$;

3) $D_1 = \neg p(x, y, a) \vee \neg p(y, z, b) \vee \neg p(x, b, v) \vee \neg p(a, z, v)$ та $D_2 = p(f(x, y), x, y)$.

5. Доведіть чи спростуйте методом резолюцій:

1) $\forall x (G(x) \to L(x)) \,\&\, \neg \exists x (K(x) \,\&\, L(x)) \,\&\, \forall x (\neg G(x) \to \neg E(x)) \models \neg \exists x (K(x) \,\&\, E(x))$;

2) $\forall x \forall y \forall z (R(x, y) \,\&\, R(y, z) \to R(x, z)) \models \forall x \forall y \forall z \forall u (R(x, y) \,\&\, R(y, z) \,\&\, R(z, u) \to R(x, u))$;

3) $\forall x \forall y (B(x, y) \to P(x, y)) \models \forall y (\forall x \neg P(x, y) \to \forall x \neg B(x, y))$;

4) $\forall x (L(x) \to T(x)) \models \forall x \forall y (L(x) \,\&\, S(x, y) \to \exists z (T(z) \,\&\, S(z, y)))$;

5) $\forall x \forall y (B(x, y) \leftrightarrow A(x) \,\&\, A(y)) \models \forall x \forall y (B(x, y) \to B(y, x))$.

3. МОДУЛЬ 3. ТЕОРІЯ АЛГОРИТМІВ

3.1. ПРАКТИЧНА РОБОТА 8. МАШИНА З НАТУРАЛЬНОЗНАЧНИМИ РЕГІСТРАМИ. МАШИНА ТЮРІНГА

ВИКОРИСТАНО МАТЕРІАЛ:

1. ***Зубенко В. В. Алгоритмічні процедури: навчальний посібник / В. В. Зубенко, О. В. Шишацька, Ф. А. Асроров. – Київ, 2023. 56 с.***
2. ***С.С.Шкільняк С.С. Теорія алгоритмів. Приклади та задачі: Навчальний посібник. – Київ: Видавничо-поліграфічний центр "Київський університет", 2012. – 144 с. (сторінка 14-26)***

Машина з натуральнозначними регістрами

Машина з натуральнозначними регістрами (МНР) є ідеалізованою моделлю комп'ютера. МНР складається із послідовності регістрів, взагалі кажучи, необмеженої, вмістом яких є натуральні числа.

Регістри нумеруємо натуральними числами, починаючи з 0, позначаючи їх $R_0, R_1, \ldots, R_n, \ldots$

Вміст регістру R_n позначаємо $'R_n$.

Послідовність ($'R_0, 'R_1, \ldots, 'R_n, \ldots$) вмістів регістрів МНР називатимемо *конфігурацією* МНР.

МНР може змінити вміст регістрів згідно виконуваної нею команди. Скінченний список команд утворює *програму* МНР. Команди програми послідовно нумеруємо натуральними числами, починаючи з 1. Номер команди в програмі називатимемо *адресою* команди.

МНР-програму з командами $I_1, I_2, \ldots, I_k$ позначимо $I_1 I_2 \ldots I_k$.

Довжину (кількість команд) МНР-програми P позначимо $|P|$.

Команди МНР бувають 4-х типів.

Тип 1. Обнулення n-го регістру $Z(n)$: $'R_n := 0$.

Тип 2. Збільшення вмісту n-го регістру на 1 $S(n)$: $'R_n := 'R_n+1$.

Тип 3. Копіювання вмісту регістру $T(m, n)$: $'R_n := 'R_m$ (при цьому $'R_m$ не змінюється).

Тип 4. Умовний перехід $J(m, n, q)$: *якщо* $'R_n = 'R_m$, *то перейти до виконання q-ї команди*, *інакше виконувати наступну за списком команду програми*.

Число q в команді $J(m, n, q)$ назвемо *адресою переходу*.

Команди типів 1–3 називають *арифметичними*. Після виконання арифметичної команди МНР повинна виконувати наступну за списком команду програми.

Виконання однієї команди МНР назвемо *кроком* МНР.

Зауважимо, що формальними моделями алгоритмів є МНР-програми, поняття МНР використовується для опису функціонування МНР-програм.

Виконання програми МНР починає, перебуваючи в деякій початковій конфігурації, з виконання першої за списком команди. Наступна для виконання команда програми визначається так, як описано вище. Виконання програми завершується (програма зупиняється), якщо наступна для виконання команда відсутня (тобто номер наступної команди перевищує номер останньої команди програми). Конфігурація МНР в момент завершення виконання програми називається фінальною, вона визначає результат роботи

МНР-програми над даною початковою конфігурацією.

Якщо МНР-програма P ніколи не зупиняється при роботі над початковою конфігурацією (a_0, a_1, ...), то цей факт позначаємо $P(a_0, a_1, ...)\uparrow$, якщо коли-небудь зупиниться, то це позначаємо $P(a_0, a_1,...)\downarrow$.

Якщо МНР-програма P при роботі над початковою конфігурацією (a_0, a_1, ...) зупиняється із фінальною конфігурацією (b_0, b_1, ...), це позначаємо $P(a_0, a_1, ...)\downarrow(b_0, b_1, ...)$.

Замість $P(a_1, a_2, ...)\downarrow(b, ...)$ надалі писатимемо $P(a_1, a_2, ...)\downarrow b$.

Кожна МНР-програма визначає відображення $N^N \to N^N$, де N^N – множина всіх нескінченних послідовностей натуральних чисел. Зрозуміло, що таке відображення однозначне. Однак МНР-програми як моделі алгоритмів є фінітними об'єктами. Кожна МНР-програма в процесі виконання використовує тільки скінченну множину регістрів, усі вони явно вказані у МНР-програмі. Тому при розгляді відображень, які задаються МНР-програмами, природно обмежитися скінченними послідовностями натуральних чисел. Таким чином, надалі розглядаємо тільки скінченні конфігурації.

Конфігурацію вигляду (a_0, a_1, ..., a_n, 0, 0, ...), в якій $'R_m = 0$ для всіх $m > n$, назвемо *скінченною.* Таку конфігурацію позначаємо (a_0, a_1, ..., a_n).

Якщо МНР-програма P починає роботу над скінченною початковою конфігурацією, то в процесі виконання P МНР перебуватиме тільки в скінченних конфігураціях.

МНР-програма P стандартна, якщо в P для кожної команди вигляду $J(m, n, q)$ виконується умова $q \le |P| + 1$.

Конкатенацією стандартних МНР-програм $P = I_1 I_2 ... I_k$ та $Q = I_1 I_2 ... I_m$ назвемо стандартну МНР-програму $I_1 ... I_k I_{k+1} ... I_{k+m}$, де команди $I_{k+1}, ..., I_{k+m}$ по суті є командами програми Q, у яких кожна команда вигляду $J(m, n, q)$ замінена командою $J(m, n, q + k)$.

МНР-програми P та Q *еквівалентні*, якщо при роботі над однаковими початковими конфігураціями вони або обидві зупиняються з однаковими фінальними конфігураціями, або обидві не зупиняються.

МНР-програма P *обчислює* часткову n-арну функцію $f : N^n \to N$, якщо

$$f(a_1, a_2, ..., a_n) = b \Leftrightarrow P(a_1, a_2, ..., a_n)\downarrow(b, ...).$$

Функція $f : N^n \to N$ *МНР-обчислювана*, якщо існує МНР-програма, яка обчислює цю функцію.

Кожна МНР-програма обчислює нескінченну кількість функцій, заданих на N, але, зафіксовуючи наперед *арність* функцій (тобто формат вхідних даних – кількість компонент початкових конфігурацій), отримуємо, що кожна МНР-програма обчислює *єдину* функцію заданої арності.

Кожну функцію, задану на N, можна трактувати як предикат, інтерпретуючи 1 та 0 як істиннісні значення T та F відповідно. Фактично в цьому випадку в ролі предиката виступає його характеристична функція.

Розглянемо приклади МНР-програм для функцій та предикатів.

***Приклад* 1.** МНР-програма для всюди невизначеної функції:

1) $J(0, 0, 1)$

***Приклад* 2.** МНР-програма для функції $f(x) = \mathrm{sg}(x)$:

1) $J(0, 1, 4)$

2) $Z(0)$

3) $S(0)$

***Приклад* 3.** МНР-програма для предиката "$x = y$":

1) $J(0, 1, 3)$

2) $J(0, 0, 4)$

3) $S(2)$

4) $T(2, 0)$

***Приклад* 4.** МНР-програма для функції $f(x, y) = x+y$:

1) $J(1, 2, 5)$

2) $S(0)$

3) $S(2)$

4) $J(0, 0, 1)$

***Приклад* 5.** МНР-програма для функції $f(x) = 3x$:

1) $T(0, 1)$

2) $J(1, 2, 6)$

3) $S(0)$

4) $S(0)$

5) $S(2)$

6) $J(0, 0, 2)$

***Приклад* 6.** МНР-програма для функції $f(x, y) = x - y$:

1) $J(0, 1, 5)$

2) $S(1)$

3) $S(2)$

4) $J(0, 0, 1)$

5) $T(2, 0)$

***Приклад* 7.** МНР-програма для функції $f(x, y) = x \dot{-} y$

1) $J(0, 1, 7)$

2) $J(0, 2, 6)$

3) $S(1)$

4) $S(2)$

5) $J(0, 0, 1)$

6) $Z(2)$

7) $T(2, 0)$

***Приклад* 8.** МНР-програма для функції $f(x, y) = \max(x, y)$:

1) $J(0, 2, 5)$

2) $J(1, 2, 6)$

3) $S(2)$

4) $J(0, 0, 1)$

5) $T(1, 0)$

***Приклад* 9.** МНР-програма для функції $f(x) = x/2$:

1) $J(0, 2, 6)$

2) $S(2)$

3) $S(2)$

4) $S(1)$

5) $J(0, 0, 1)$

6) $T(1, 0)$

***Приклад* 10.** МНР-програма для функції $f(x) = [x/2]$:

1) $J(0, 2, 7)$

2) $S(2)$

3) $J(0, 2, 7)$

4) $S(2)$

5) $S(1)$

6) $J(0, 0, 1)$

7) $T(1, 0)$

***Приклад* 11.** МНР-програма для функції $f(x, y, z) = x + \min(y, z)$:

1) $J(0, 3, 6)$

2) $J(2, 3, 6)$

3) $S(3)$

4) $S(0)$

5) $J(0, 0, 1)$

***Приклад* 12.** МНР-програма для функції $f(x, y, z) = \max(x + y, z)$:

1) $J(1, 3, 5)$

2) $S(3)$

3) $S(0)$

4) $J(0, 0, 1)$

5) $J(0, 4, 9)$

6) $J(2, 4, 10)$

7) $S(4)$

8) $J(0, 0, 5)$

9) $T(1, 0)$

***Приклад* 13.** МНР-програма для функції $f(x, y) = x \cdot y$,

1) $J(3, 1, 9)$

2) $J(0, 2, 6)$

3) $S(2)$

4) $S(4)$

5) $J(0, 0, 2)$

6) $Z(2)$

7) $S(3)$

8) $J(0, 0, 1)$

9) $T(4, 0)$

ВПРАВИ

1. Вкажіть МНР-програми для функцій:

1) $f(x, y) = 3x+y+1$; 2) $f(x, y) = x+2y+2$;

3) $f(x) = (x+1)/3$; 4) $f(x) = [x/3]$;

5) $f(x, y) = 2x-y$; 6) $f(x, y) = 3y-x$;

7) $f(x, y, z) = (x-y)+z$; 8) $f(x, y, z) = x-(y+z)$;

9) $f(x) = \mathrm{nsg}(x)$; 10) $f(x, y) = \mathrm{sg}(x+y)$;

11) $f(x, y) = \mathrm{sg}(x\cdot y)$; 12) $f(x, y, z) = \mathrm{nsg}(x+y+z)$;

13) $f(x, y) = \min(x, y)$; 14) $f(x, y) = \max(2x, y)$;

15) $f(x, y, z) = \max(x, y)+z$; 16) $f(x, y, z) = x-\min(y, z)$;

17) $f(x, y, z) = \min(x+y, z)$; 18) $f(x, y, z) = \max(x, y-z)$;

19) $f(x, y) = \max(x, 3y)+2x$; 20) $f(x, y) = 3x-\min(2y, x)$;

21) $f(x) = x!$; 22) $f(x, y) = x^y$.

2. Вкажіть МНР-програми для предикатів:

1) "x – непарне число"; 2) "x – парне число";

3) "$x \geq y$";. 4) "$x \neq y$";

5) "$x > y$"; 6) "$x < y$".

Машини Тьюрінга

Під *машиною Тьюрінга* (скор. МТ) розуміємо впорядковану 5-ку (Q, T, δ, q_0, q^*), де:

– Q – скінченна множина внутрішніх станів;

– T – скінченний алфавіт символів стрічки; тут T містить спеціальний символ порожньої клітки λ;

– $\delta : Q\times T\to Q\times T\times\{R, L, \varepsilon\}$ – однозначна функція переходів;

– $q_0\in Q$ – початковий стан;

– $q^*\in Q$ – фінальний стан.

МТ *детермінована*, якщо функція δ однозначна, інакше МТ *недетермінована*.

Функцію переходів задаватимемо скінченною множиною команд одного з 3-х видів: $qa\to pbR$, $qa\to pbL$ та $qa\to pb$, де $p, q\in Q$, $a, b\in T$, $\to\notin Q\cup T$. При цьому, зазвичай, не для всіх пар $(q, a)\in Q\times T$ існує команда з лівою частиною qa. Це означає, що функція δ не є тотальною. Проте зручніше вважати функцію δ тотальною, тому для всіх пар $(q, a)\notin D_\delta$ неявно (не додаючи відповідні команди вигляду $qa\to qa$) вводимо таке довизначення: $\delta(q, a) = (q, a, \varepsilon)$.

Неформально МТ складається з скінченної пам'яті, розділеної на клітки нескінченної з обох боків стрічки та голівки читання-запису. В кожній клітці стрічки міститься єдиний символ із T, причому в кожен даний момент стрічка містить скінченну кількість символів, відмінних від символа λ. Голівка читання-запису в кожен даний момент оглядає єдину клітку стрічки. Якщо МТ знаходиться в стані q та голівка читає символ a, то при виконанні команди $qa\to pbR$ (команди $qa\to pbL$, $qa\to pb$) МТ переходить в стан p, замість символу a записує на стрічці символ b та зміщує голівку на 1 клітку направо (відповідно на 1 клітку наліво, залишає голівку на місці).

Конфігурація, або *повний стан МТ* – це слово вигляду xqy, де x, $y \in T^*$, $q \in Q$. Неформально це означає, що на стрічці записане слово xy, тобто зліва і справа від xy можуть стояти тільки символи λ, МТ знаходиться в стані q, голівка читає 1-й символ підслова y.

Конфігурацію вигляду q_0x, де 1-й та останній символи слова x відмінні від λ, назвемо *початковою*.

Конфігурацію вигляду xq^*y назвемо *фінальною*.

Після переходу до фінального стану, отже, до фінальної конфігурації, МТ зупиняється.

Нехай МТ знаходиться в конфігурації $xcqay$, де $x, y \in T^*$, $a, c \in T$, $q \in Q$. Після виконання команди $qa \to pbR$ (команди $qa \to pbL$, команди $qa \to pb$) МТ перейде до конфігурації $xcbpy$ (відповідно до конфігурації $xpcby$, конфігурації $xcpby$).

Кожна МТ задає вербальне відображення $T^* \to T^*$ таким чином.

МТ M переводить слово $u \in T$ в слово $v \in T^*$ (цей факт записуємо $v = M(u)$), якщо вона з початкової конфігурації q_0u переходить до фінальної конфігурації xq^*y, де $xy = \alpha v \beta$, $\alpha, \beta \in \{\lambda\}^*$. При цьому перший та останній символи слова v відмінні від λ, або $v = \varepsilon$.

Якщо МТ M, починаючи роботу з початкової конфігурації q_0u, ніколи не зупиниться, кажуть, що M *зациклюється* при роботі над словом u. Тоді $M(u)$ не визначене.

МТ M_1 та M_2 *еквівалентні*, якщо вони задають одне і те ж вербальне відображення.

МТ M *обчислює* часткову функцію $f: N^k \to N$, якщо вона кожне слово вигляду $|^{x_1}\#|^{x_2}\#\ldots\#|^{x_k}$ переводить в слово $|^{f(x_1,\ldots,x_k)}$ у випадку $(x_1, \ldots, x_k) \in D_f$, та $M(|^{x_1}\#|^{x_2}\#\ldots\#|^{x_k})$ невизначене при $(x_1, \ldots, x_k) \notin D_f$.

Функція *обчислювана за Тьюрінгом*, або *МТ-обчислювана*, якщо існує МТ, яка її обчислює.

Кожна МТ обчислює нескінченну множину функцій натуральних аргументів та значень, але зафіксовуючи наперед арність функцій, дістаємо, що кожна МТ обчислює *єдину* функцію заданої арності.

Розглянемо приклади МТ.

***Приклад* 1.** МТ, яка обчислює функцію $x+y$:

$q_0| \to q_0|R$

$q_0\# \to q_0|R$

$q_0\lambda \to q_1\lambda L$

$q_1| \to q^*\lambda$

***Приклад* 2.** МТ, яка обчислює функцію $f(x) = \mathrm{sg}(x)$:

$q_0\lambda \to q^*\lambda$

$q_0| \to q_1|R$

$q_1| \to q_1\lambda R$

$q_1\lambda \to q^*\lambda$

***Приклад* 3.** МТ, яка обчислює предикат "x парне":

$q_0| \to q_1\lambda R$

$q_1| \to q_0\lambda R$

$q_0\lambda \to q^*|$

$q_1\lambda \to q^*\lambda$

***Приклад* 4.** МТ, яка обчислює предикат "$x = 2$":

$q_0\lambda \to q^*\lambda$ $(x = 0)$

$q_0 | \to q_1\lambda R$ $(x \geq 1)$

$q_1\lambda \to q^*\lambda$ $(x = 1)$

$q_1 | \to q_2\lambda R$ $(x \geq 2)$

$q_2\lambda \to q^*|$ $(x = 2)$

$q_2 | \to q_3\lambda R$

$q_3 | \to q_3\lambda R$

$q_3\lambda \to q^*\lambda$

***Приклад* 5.** МТ, яка обчислює функцію $f(x, y) = x - y$:

$q_0 | \to q_1\lambda R$

$q_1| \to q_1 | R$

$q_1\# \to q_1\#R$

$q_1\lambda \to q_2\lambda L$

$q_2 | \to q_3\lambda L$

$q_3 | \to q_3|L$

$q_3 \# \to q_3\#L$

$q_3\lambda \to q_0\lambda R$

$q_{2v}\# \to q^*|$

$q_0 \# \to q_4\lambda R$

$q_4\lambda \to q^*\lambda$

***Приклад* 6.** МТ, яка обчислює функцію $f(x, y) = x \dot{-} y$

$q_0 | \to q_1\lambda R$

$q_1| \to q_1 | R$

$q_1\# \to q_1\#R$

$q_1\lambda \to q_2\lambda L$

$q_2| \to q_3\lambda L$

$q_3| \to q_3 | L$

$q_3\# \to q_3 \#L$

$q_3\lambda \to q_0\lambda R$

$q_2 \# \to q^*|$

$q_0 \# \to q_4\lambda R$

$q_4| \to q_4\lambda R$ (єдина відмінність від МТ для $f(x, y) = x - y$)

$q_4\lambda \to q^*\lambda$

***Приклад* 7.** МТ, яка обчислює функцію $f(x, y) = x+2y$:

$q_0 | \to q_0 | R$

$q_0\# \to q_0\#R$

$q_0\lambda \to q_1\lambda L$

$q_1 | \to q_2\lambda R$

$q_2 | \to q_2 | R$

$q_2\lambda \to q_3 | L$

$q_3 | \to q_3 | L$

$q_3\lambda \to q_1 | L$

$q_1 \# \to q_4 | L$

$q_4 | \to q_4 | L$

$q_4\lambda \to q_5\lambda R$

$q_5 | \to q^*\lambda$

***Приклад* 8.** МТ, яка обчислює функцію $f(x) = [x/3]$:

$q_0\lambda \to q^*\lambda$

$q_0 | \to q_1\lambda R$

$q_1 | \to q_2\lambda R$

$q_2 | \to q_2 | R$

$q_2\lambda \to q_3\lambda L$

$q_2 a \to q_3 aL$

$q_3 | \to q_4 aL$

$q_4 | \to q_4 | L$

$q_4\lambda \to q_0\lambda R$

$q_1 a \to q_0 a$

$q_3 \lambda \to q_0 \lambda R$

$q_0 a \to q_0 | R$

***Приклад* 9.** МТ, яка обчислює функцію $f(x, y) = x \cdot y$

$q_0 \# \to q_1\lambda R$

$q_1 | \to q_1\lambda R$

$q_1\lambda \to q^*\lambda$

$q_1 a \to q_1 | R$

$q_0 | \to q_2\lambda R$

$q_2 | \to q_2 | R$

$q_2 \# \to q_3 \# R$

$q_3 | \to q_4\lambda R$

$q_4 | \to q_4 | R$

$q_4 a \to q_4 aR$

$q_4\lambda \to q_5 aL$

$q_5 | \to q_5 | L$

$q_5 a \to q_5 aL$

$q_5\lambda \to q_3|R$

$q_3a \to q_6\, aL$

$q_6\,| \to q_6\,|\,L$

$q_6\# \to q_6\,\#L$

$q_6\lambda \to q_0\lambda R$

$q_3\lambda \to q_7\lambda L$

$q_7\# \to q_7\lambda L$

$q_7| \to q_7\lambda L$

$q_7\lambda \to q^*\lambda$

***Приклад* 10.** МТ, яка обчислює функцію $f(x) = 2^x$

$q_0\,| \to q_0\,|\,R$

$q_0\lambda \to q_1aL$

$q_1\,| \to q_1\,|\,L$

$q_1\lambda \to q_2\lambda R$

$q_2\,| \to q_3\lambda R$

$q_3\,| \to q_3\,|\,R$

$q_3a \to q_3\, aR$

$q_3\lambda \to q_4\lambda L$

$q_4a \to q_5\lambda R$

$q_5a \to q_5\, aR$

$q_5\lambda \to q_6\, aL$

$q_6a \to q_6\, aL$

$q_6\lambda \to q_4aL$

$q_4| \to q_4\,|\,L$

$q_4\lambda \to q_2\lambda R$

$q_2a \to q_2\,|\,R$

$q_2\lambda \to q^*\lambda$

***Приклад* 11.** МТ, яка кожне слово $x \in T^*$ переводить в слово $x\#x$ (тут $\# \notin T$).

$q_0\, t \to q_0\, tR$ для всіх $t \in T$

$q_0\lambda \to q_1\#L$

$q_1\, t \to q_1\, tL$ для всіх $t \in T$

$q_1\lambda \to q_2\lambda R$

$q_2\, t \to q_t\, \lambda R$ для всіх $t \in T$

$q_t\, p \to q_t\, pR$ для всіх $t \in T$, $p \in T \cup \{\#\}$

$q_t\, \lambda \to q'_t\, tL$ для всіх $t \in T$

$q'_t\, p \to q'_t\, pL$ для всіх $t \in T$, $p \in T \cup \{\#\}$

$q'_t\, \lambda \to q_2tR$ для всіх $t \in T$

$q_2\# \to q^*\#$

ВПРАВИ

1. Вкажіть МТ, яка кожне слово $x \in T^*$ переводить:

1) в слово xx;

2) в слово xx^R; де x^R – дзеркальне відображення слова x;

3) в слово x^R; де x^R – дзеркальне відображення слова x.

2. Вкажіть МТ для функцій:

1) $f(x) = \mathrm{sg}(x/2)$; 2) $f(x) = \mathrm{nsg}[x/2]$;

3) $f(x, y) = \mathrm{sg}(x+y)$; 4) $f(x, y) = \mathrm{nsg}(x \cdot y)$;

5) $f(x) = x/3$; 6) $f(x) = [x/2]$;

7) $f(x, y) = (x+1) - y$; 8) $f(x,y) = (x+2) \dot{-} y$;

9) $f(x,y) = x \dot{-} (y+1)$; 10) $f(x, y) = |x - (y+2)|$;

11) $f(x, y) = 3x+y$; 12) $f(x, y) = x - 2y$;

13) $f(x, y, z) = 2x+y+z$; 14) $f(x, y, z) = x+y+3z$;

15) $f(x, y, z) = (x+y) - z$; 16) $f(x, y, z) = 2x - (y+z)$;

17) $f(x, y) = \max(x, y)$; 18) $f(x, y) = \min(x, y)$;

19) $f(x, y) = [x/y]$; 20) $f(x, y) = \mathrm{mod}(x, y)$;

21) $f(x) = x!$; 22) $f(x, y) = x^y$.

3. Вкажіть МТ для предикатів:

1) "x непарне"; 2) "$x = 1$";

3) "$x > y$"; 4) "$x < y$";

5) "$x \geq y$"; 6) "$x \neq y$".

7) "$x \leq y$"; 8) "$x \neq 3$".

3.2. ПРАКТИЧНА РОБОТА 9. СИСТЕМА ПОСТА

ВИКОРИСТАНО МАТЕРІАЛ:

1. ***Зубенко В. В. Алгоритмічні процедури: навчальний посібник / В. В. Зубенко, О. В. Шишацька, Ф. А. Асроров. – Київ, 2023. 56 с.***
2. ***С.С.Шкільняк С.С. Теорія алгоритмів. Приклади та задачі: Навчальний посібник. – Київ: Видавничо-поліграфічний центр "Київський університет", 2012. – 144 с. (сторінка 32-36)***

Системи Поста

Канонічною системою Поста над алфавітом T називають формальну систему (T^*, A, P), в якої множина аксіом A є скінченною підмножиною множини T^*, а множина правил виведення P складається зі слів вигляду $\alpha_0 S_1 \alpha_1 \ldots \alpha_{m-1} S_m \alpha_m \to \beta_0 S_{j_1} \beta_1 \ldots \beta_{n-1} S_{j_n} \beta_n$. Тут $\to \notin T$, всі α_k та β_i – фіксовані слова із T^* , всі символи $S_k \notin T$, причому всі $j_i \in \{1, \ldots, m\}$.

Символи S_k призначені для позначення довільних слів із T^*.

Системи Поста звичайно позначають у вигляді $P = (T, A, P)$.

Множина правил P визначає на словах із T^* відношення *безпосереднього* виведення таким чином: $\sigma \Rightarrow_P \tau$, якщо існує правило $\alpha_0 S_1 \alpha_1 ... \alpha_{m-1} S_m \alpha_m \to \beta_0 S_{j_1} \beta_1 .. \beta_{n-1} S_{j_n} \beta_n \in P$ таке, що для деяких слів $\varphi_1, ..., \varphi_m \in T^*$ маємо $\sigma = \alpha_0 \varphi_1 \alpha_1 ... \varphi_m \alpha_m$, $\tau = \beta_0 \varphi_{j_1} \beta_1 .. \beta_{n-1} \varphi_{j_n} \beta_n$.

Рефлексивно-транзитивне замикання відношення $\Rightarrow_P$ позначаємо $\overset{*}{\Rightarrow}_P$. Інакше кажучи, $\sigma \overset{*}{\Rightarrow}_P \tau$ означає, що слово τ отримане із слова σ за допомогою скінченної кількості застосувань правил із P.

Слово τ породжується системою Поста P, якщо $\alpha \overset{*}{\Rightarrow}_P \tau$ для деякої $\alpha \in A$. Цей факт записуємо $_P|\!-\tau$ та називаємо таке слово τ *теоремою* системи Поста P.

Множину $Th(P) = \{\tau \in T^* \mid {}_P|\!-\tau\}$ називатимемо *множиною теорем* системи Поста P.

Для завдання системи Поста достатньо вкажіть множину правил та множину аксіом.

У випадку необхідності вказуємо і алфавіт T.

***Приклад* 1.** Система Поста із $A = \{a, b, \varepsilon\}$ та $P = \{S \to aSa,\ S \to bSb\}$ породжує всі слова-паліндроми в алфавіті $\{a, b\}$, тобто слова, які читаються однаково зліва направо і справа наліво.

Множина $X \subseteq T^*$ *породжувана за Постом*, якщо існують алфавіт $T' \supseteq T$ та система Поста $P = (T', A, P)$ такі, що $Th(P) \cap (T^*) = X$.

Функція $f : N^k \to N$ *обчислювана за Постом*, якщо множина $\{|^{x_1}\#|^{x_2}\#\ldots\#|^{x_k}\#|^{f(x_1,\ldots,x_k)} \mid (x_1,\ldots, x_k) \in D_f\}$. породжувана за Постом.

Таким чином, обчислюваність функцій за Постом – це породжуваність за Постом графіків таких функцій.

Наведемо приклади функцій і предикатів, обчислюваних за Постом.

***Приклад* 2.** Система Поста для функції $f(x, y) = x+y$:

$A = \{\#\#\}$;

$P = \{X\#Y\#R \to X|\#Y\#R|,$

$X\#Y\#R \to X\#Y|\#R|\ \}$.

***Приклад* 3.** Система Поста для функції $f(x, y) = x - y$:

$A = \{\#\#\}$;

$P = \{X\#Y\#R \to X|\#Y\#R|,$

$X\#Y\#R| \to X\#Y|\#R\ \}$.

***Приклад* 4.** Ще одна система Поста для функції $f(x, y) = x - y$:

$A = \{\#\#\}$;

$P = \{X\#Y\#R \to X|\#Y|\#R,$

$X\#\#R \to X|\#\#R|\ \}$.

***Приклад* 5.** Система Поста для функції $f(x, y) = x \dot{-} y$:

$A = \{\#\#\}$;

$P = \{X\#Y\#R \to X|\#Y|\#R,$

$X\#\#R \to X|\#\#R|,$

$\#Y\# \to \#Y|\#\ \}$.

***Приклад* 6.** Система Поста для функції $f(x, y) = |x - y|$:

$A = \{\#\#\}$;

$P = \{X\#Y\#R \to X|\#Y|\#R,$

$X\#\#R \to X|\#\#R|,$

$X\#Y\#R \to Y\#X\#R\ \}$.

***Приклад* 7.** Система Поста для функції $f(x, y) = x \cdot y$:

$A = \{\#\#\}$;

$P = \{X\#Y\#R \to X|\#Y\#RY,$

$X\#Y\#R \to X\#Y|\#RX\ \}$.

***Приклад* 8.** Система Поста для функції $f(xy) = x^2$:

$A = \{\#|\}$;

$P = \{X\#R \to X|\#RXX|\ \}$.

***Приклад* 9.** Система Поста для функції $f(x) = 2^x$:

$A = \{\#\}$;

$P = \{X\#R \to X|\#RR\ \}$.

***Приклад* 10.** Система Поста для функції $f(x, y) = \max(x, y)$:

$A = \{\#\#\}$;

$P = \{X\#X\#X \to X|\#X|\#X|,$

$X\#XS\#XS \to X\#XS|\#XS|,$

$XS\#X\#XS \to XS|\#X\#XS|\ \}$.

***Приклад* 11.** Система Поста для функції $f(x, y) = x + 2^y$:

$A = \{\#\#\ |\}$;

$P = \{X\#Y\#XS \to X|\#Y\#XS|,$

$X\#Y\#XS \to X\#Y|\#XSS\ \}$.

***Приклад* 12.** Система Поста для функції $f(x, y) = x - y^2$:

$A = \{\#\#\}$;

$P = \{X\#\#R \to X|\#\#R|,$

$X\#Y\#RYY| \to X\#Y|\#R\ \}$.

***Приклад* 13.** Система Поста для функції $f(x) = x^2 - 2x$:

$A = \{\#,\ ||\#\}$;

$P = \{X|\#R \to X||\#RXX|\ \}$.

В цьому прикладі треба врахувати, що $f(1)\uparrow$.

***Приклад* 14.** Система Поста для функції $f(x) = x^3$:

$A = \{**\}$;

$P = \{X*Q*R \to X|*QXX|*RQQQXXX|,$

$X*Q*R \to X\#R\ \}$.

Для обчислення значення $f(x+1) = (x+1)^3 = f(x) + 3x^2 + 3x + 1$ потрібне x^2, тому теоремами СП мусять також бути коди трійок (x, x^2, x^3). Але теоремами в алфавіті {|, #} можуть бути тільки коди елементів графіка $f(x)$, тому для кодування трійок (x, x^2, x^3) використано ∗.

***Приклад* 15.** Система Поста для функції $f(x) = x!$:

$A = \{\#|\}$;

$P = \{X\#F \to X|*F***,$

$S*F*A*B*M \to S*F*A|*B*MB,$

$S*F*A*B*M \to S*F*A*B|*MA,$

$S*F*S*F*M \to S\#M\ \}$.

До теорем СП належать коди 5-рок (x+1, x!, a, b, ab), для їх кодування використано ∗.

Із кодів 5-рок (x+1, x!, x+1, x!, $(x+1)\cdot x!$) можна вивести закодовані в алфавіті {|, #} коди пар (x+1, $(x+1)!$).

***Приклад* 16.** Система Поста для функції $f(x) = [\sqrt{x}]$:

$A = \{**\}$;

$P = \{X** \to X|**,$

$QS|*Q*R \to QS|*QRR|*R|,$

$X*X*R \to X\#R,$

$X*XS|*R| \to X\#R\ \}$.

До теорем СП належать коди трійок (x, r^2, r), для їх кодування використано ∗.

Із кодів трійок (x, x, r) та (x, $(r+1)^2$, r+1) при умові $r^2 \le x < (r+1)^2$ можна вивести закодовані в алфавіті {|, #} коди пар (x, r).

***Приклад* 17.** Система Поста для предиката "$x = y$":

$A = \{\#\#\ |\}$;

$P = \{X\#Y\#R \to X|\#Y|\#R,$

$X\#\#R \to X|\#\#,$

$\#Y\#R \to \#Y|\#\ \}$.

ВПРАВИ

1. Вкажіть системи Поста для предикатів:

1) "x парне"; 2) "$x \ne y$";

3) "$x > y$"; 4) "$x \le y$".

2. Вкажіть системи Поста для функцій:

1) $f(x) = \mathrm{nsg}(x)$; 2) $f(x) = \mathrm{sg}(x)$;

3) $f(x, y) = \min(x, y)$; 4) $f(x, y) = \max(x+1, y)$;

5) $f(x, y, z) = (2x+y)\cdot z$; 6) $f(x) = 3^x + x$;

7) $f(x) = x^3 + 5x$; 8) $f(x) = x^4 - x$;

9) $f(x) = x^4 - 2x$; 10) $f(x) = x^3 - 3x$;

11) $f(x) = 3^x - 4x$; 12) $f(x) = 2^x - 5x$;

13) $f(x, y) = 3^x + 2^y$;

14) $f(x, y) = 2^x - 3^y$;

15) $f(x, y, z) = x \cdot y \cdot z$;

16) $f(x, y, z) = x \cdot y \cdot (z+1)$;

17) $f(x, y) = x \cdot y^2$;

18) $f(x, y) = x^2 \cdot (y+1)$;

19) $f(x, y) = (x+1)^2 \cdot y$;

20) $f(x, y) = 2^x \cdot y$;

21) $f(x, y) = (x+1) \cdot 2^y$;

22) $f(x, y) = 3^x \cdot (y+1)$;

23) $f(x) = [\log_2 x]$;

24) $f(x) = [x\sqrt{2}]$.

25) $f(x, y) = x^y$;

26) $f(x, y) = [\sqrt[x]{y}]$.

3.3. ПРАКТИЧНА РОБОТА 10. ОБЧИСЛЮВАЛЬНІСТЬ ФУНКЦІЙ. ОБЧИСЛЮВАЛЬНІ ТЕРМИ

ВИКОРИСТАНО МАТЕРІАЛ:

1. ***Зубенко В. В. Алгоритмічні процедури: навчальний посібник / В. В. Зубенко, О. В. Шишацька, Ф. А. Асроров. – Київ, 2023. 56 с.***
2. ***С.С.Шкільняк С.С. Теорія алгоритмів. Приклади та задачі: Навчальний посібник. – Київ: Видавничо-поліграфічний центр "Київський університет", 2012. – 144 с. (сторінка 61-71)***

Обчислюваність *n*-арних функцій на *N*

Основними обчислюваними операціями для *n*-арних функцій на множині *N* є наступні операції: *операція суперпозиції* $\mathbf{S}^{n+1}$, *операція примітивної рекурсії* **R**, *операція мінімізації* **M**.

Операція $\mathbf{S}^{n+1}$ *n*-арній функції $g(x_1, ..., x_n)$ та *n* функціям $g_1(x_1, ..., x_m)$, ..., $g_n(x_1, ..., x_m)$ одної і тої ж арності зіставляє функцію $f(x_1, ..., x_m) = g(g_1(x_1, ..., x_m), ..., g_n(x_1, ..., x_m))$.

Таку функцію будемо позначати $\mathbf{S}^{n+1}(g, g_1, ..., g_n)$, її арність співпадає з арністю $g_1, ..., g_n$.

Операція *примітивної рекурсії* **R** *n*-арній функції *g* та (*n*+2)-арній функції *h* зіставляє (*n*+1)-арну функцію *f*, яку позначають $\mathbf{R}(g, h)$, що задається рекурсивним визначенням:

$f(x_1, ..., x_n, 0) = g(x_1, ..., x_n)$

$f(x_1, ..., x_n, y+1) = h(x_1, ..., x_n, y, f(x_1, ..., x_n, y))$

Це означає, що для всіх $a_1, ..., a_n, b$ значення $f(a_1, ..., b_n, b)$ обчислюється так:

$f(a_1, ..., a_n, 0) = g(a_1, ..., a_n)$

$f(a_1, ..., a_n, 1) = h(a_1, ..., a_n, 0, f(a_1, ..., a_n, 0))$

... ...

$f(a_1, ..., a_n, b) = h(a_1, ..., a_n, b-1, f(a_1, ..., a_n, b-1))$

При $n = 0$ вважаємо, що функція *g* – 1-арна константа.

Операція мінімізації **M** кожній (*n*+1)-арній функції *g* зіставляє *n*-арну функцію *f*, яку позначають $\mathbf{M}(g)$, що задається співвідношенням $f(x_1, ..., x_n) = \mu_y(g(x_1, ..., x_n, y) = 0)$

Для всіх $x_1, ..., x_n$ значення $f(x_1,..., x_n)$ обчислюється так. Послідовно обчислюємо $g(x_1,..., x_n, y)$ для $y = 0$, 1, 2, ... Перше таке значення *y*, для якого $g(x_1, ..., x_n, y) = 0$, – шукане $f(x_1, ..., x_n)$. При цьому для всіх $t < y$ необхідно $g(x_1, ..., x_n, t)\downarrow \neq 0$.

Зрозуміло, що функція g може бути тотальною, а $f = \mathbf{M}(g)$ – навіть всюди невизначеною. Наприклад, $f(x) = \mu_y(x+y+1 = 0)$.

Базові функції – це найпростіші функції $\boldsymbol{o}(x) = 0$, $\boldsymbol{s}(x) = x+1$ та функції-селектори $\boldsymbol{I}_m^n(x_1, \ldots, x_n) = x_m$, де $n \geq m \geq 1$.

Всі базові функції тотальні та алгоритмічно обчислювані.

Функцію, яку можна отримати із базових функцій за допомогою скінченної кількості застосувань операцій суперпозиції та примітивної рекурсії, називають *примітивно рекурсивною функцією* (ПРФ).

Функцію, яку можна отримати із базових функцій за допомогою скінченної кількості застосувань операцій суперпозиції, примітивної рекурсії та мінімізації, називають *частково рекурсивною* функцією (ЧРФ).

Тотальну ЧРФ називають *рекурсивною* функцією (РФ).

Із наведених визначень випливають наступні твердження:

1. *Якщо функції* $g, g_1, \ldots, g_n$ *тотальні алгоритмічно обчислювані, то функція* $\mathbf{S}^{n+1}(g, g_1, \ldots, g_n)$ *тотальна алгоритмічно обчислювана*.

2. *Якщо функції* g *та* h *тотальні алгоритмічно обчислювані, то функція* $\mathbf{R}(g, h)$ *тотальна алгоритмічно обчислювана*.

3. *Якщо функція* g *алгоритмічно обчислювана, то функція* $\mathbf{M}(g)$ *алгоритмічно обчислювана*.

4. *Кожна ЧРФ – алгоритмічно обчислювана функція.*

5. *Кожна РФ та кожна ПРФ – тотальна алгоритмічно обчислювана функція.*

6. *Для класів функцій мають місце співвідношення*

$$ПРФ \subseteq РФ \subseteq ЧРФ.$$

Алгебра $(\mathfrak{R}; \mathbf{R}, \mathbf{M}, \mathbf{S}^2, \mathbf{S}^3, \ldots)$, носієм $\mathfrak{R}$ якої є клас всіх ЧРФ, а операціями – $\mathbf{R}$, $\mathbf{M}$ та $\mathbf{S}^{n+1}$, де $n \geq 1$, називається *алгеброю ЧРФ*, або *алгеброю Чорча*.

Алгебра $(\mathfrak{R}_{pr}; \mathbf{R}, \mathbf{S}^2, \mathbf{S}^3, \ldots)$, носієм $\mathfrak{R}_{pr}$ якої є клас всіх ПРФ, а операціями – $\mathbf{R}$ та $\mathbf{S}^{n+1}$, де $n \geq 1$, називається *алгеброю ПРФ*.

Введемо поняття *операторного терма алгебри ЧРФ* та *операторного терма алгебри ПРФ*.

Алфавіт складається із символів базових функцій $\boldsymbol{o}$, $\boldsymbol{s}$ та $\boldsymbol{I}_m^n$, де $n \geq m \geq 1$, символів операцій $\mathbf{R}$, $\mathbf{M}$ та $\mathbf{S}^{n+1}$, де $n \geq 1$, а також допоміжних символів "(", ")" та "," .

Індуктивне визначення ОТ алгебри ЧРФ таке:

1) кожен символ базової функції є ОТ; такі ОТ назвемо атомарними;

2) якщо $t_0, t_1, \ldots, t_n$ – ОТ, то $\mathbf{S}^{n+1}(t_0, t_1, \ldots, t_n)$ – ОТ;

3) якщо t_1 та t_2 – ОТ, то $\mathbf{R}(t_1, t_2)$ – ОТ;

4) якщо t – ОТ, то $\mathbf{M}(t)$ – ОТ.

Аналогічно дається індуктивне визначення ОТ алгебри ПРФ.

Кожна ЧРФ є значенням деякого ОТ алгебри ЧРФ. Водночас не кожен ОТ алгебри ЧРФ має певне значення. Наприклад, ОТ $\mathbf{R}(\boldsymbol{o}, \boldsymbol{I}_2^4)$ та $\mathbf{S}^3(\boldsymbol{I}_1^2, \boldsymbol{I}_2^3, \boldsymbol{I}_2^2)$ не мають значення.

Задання ЧРФ операторними термами не є однозначним. Наприклад, ОТ $\boldsymbol{o}$, $\mathbf{S}^2(\boldsymbol{o}, \boldsymbol{s})$, $\mathbf{S}^2(\boldsymbol{o}, \boldsymbol{o})$ та $\mathbf{S}^2(\boldsymbol{o}, \mathbf{S}^2(\boldsymbol{o}, \boldsymbol{s}))$ задають одну і ту ж функцію $\boldsymbol{o}(x)$.

Розглянемо приклади ПРФ, ЧРФ та РФ.

***Приклад* 1.** Функції-константи – ПРФ.

n-арна нуль-функція $\boldsymbol{o}^n(x_1, ..., x_n) = 0$ задається ОТ $\mathbf{S}^2(\boldsymbol{o}, \boldsymbol{I}_1^n)$;

n-арна константа $\boldsymbol{k}^n(x_1, ..., x_n) = k$ задається ОТ $\mathbf{S}^2(\boldsymbol{s}, \mathbf{S}^2(\boldsymbol{s},...,\mathbf{S}^2(\boldsymbol{o}, \boldsymbol{I}_1^n)...)$.

***Приклад* 2.** Функція $f(x_1, x_2) = x_1+x_2$ – ПРФ. Справді, маємо

$f(x_1, 0) = x_1 = \boldsymbol{I}_1^1(x_1)$;

$f(x_1, x_2+1) = x_1+(x_2+1) = (x_1+x_2)+1 = \boldsymbol{s}(x_1+x_2) = \boldsymbol{s}(f(x_1, x_2))$.

Таким чином, функція $f(x_1, x_2) = x_1+x_2$ виникає примітивною рекурсією із функцій $g(x_1) = \boldsymbol{I}_1^1(x_1)$ та $h(x_1, x_2, x_3) = x_3+1 = \boldsymbol{s}(x_3) = = \mathbf{S}^2(\boldsymbol{s}, \boldsymbol{I}_3^3)(x_1, x_2, x_3)$.

Операторний терм функції x_1+x_2 має вигляд $\mathbf{R}(\boldsymbol{I}_1^1, \mathbf{S}^2(\boldsymbol{s}, \boldsymbol{I}_3^3))$.

***Приклад* 3.** Функція $f(x_1, x_2) = x_1 \cdot x_2$ – ПРФ. Справді, маємо

$f(x_1, 0) = 0 = \boldsymbol{o}(x_1)$;

$f(x_1, x_2+1) = x_1 \cdot (x_2+1) = x_1 \cdot x_2+x_1 = f(x_1, x_2)+x_1$.

Отже, $f(x_1, x_2) = x_1 \cdot x_2$ виникає примітивною рекурсією із функцій $g(x_1) = \boldsymbol{o}(x_1)$ та $h(x_1, x_2, x_3) = x_3+x_1$. За прикладом 2 функція h є ПРФ, тому f – ПРФ. Операторний терм $\otimes$ функції $x_1 \cdot x_2$ має вигляд $\mathbf{R}(\boldsymbol{o}, \mathbf{S}^3(\oplus, \boldsymbol{I}_3^3, \boldsymbol{I}_1^3))$, де $\oplus$ – операторний терм функції x_1+x_2.

***Приклад* 4.** Функція $f(x_1, x_2) = (x_1)^{x_2}$ – ПРФ. Справді, маємо

$f(x_1, 0) = (x_1)^0 = 1$;

$f(x_1, x_2+1) = (x_1)^{x_2+1} = (x_1)^{x_2} \cdot x_1 = f(x_1, x_2) \cdot x_1$.

Таким чином, функція $f(x_1, x_2) = (x_1)^{x_2}$ виникає примітивною рекурсією із функцій $g(x_1) = 1^1(x_1)$ та $h(x_1, x_2, x_3) = x_3 \cdot x_1$. За прикладом 3 функція h є ПРФ, тому f – ПРФ.

ОТ функції $(x_1)^{x_2}$ має вигляд $\mathbf{R}(\mathbf{S}^2(\boldsymbol{s}, \boldsymbol{o}), \mathbf{S}^3(\otimes, \boldsymbol{I}_3^3, \boldsymbol{I}_1^3))$, де $\otimes$ –ОТ функції $x_1 \cdot x_2$.

***Приклад* 5.** Функція $sg(x_1) = \begin{cases} 0, \text{ якщо } x_1 = 0, \\ 1, \text{ якщо } x_1 \geq 1, \end{cases}$ – ПРФ.

Справді, маємо: $sg(0) = 0 = \boldsymbol{o}(x_1)$; $sg(x_1+1) = 1$.

ОТ функції $sg(x_1)$ має вигляд $\mathbf{R}(\boldsymbol{o}, \mathbf{S}^2(\boldsymbol{s}, \mathbf{S}^2(\boldsymbol{o}, \boldsymbol{I}_1^2)))$.

***Приклад* 6.** Функція $nsg(x_1) = \begin{cases} 1, \text{ якщо } x_1 = 0, \\ 0, \text{ якщо } x_1 \geq 1, \end{cases}$ – ПРФ.

Справді, маємо: $nsg(0) = 1 = \boldsymbol{s}(\boldsymbol{o}(x_1))$; $nsg(x_1+1) = 0$.

ОТ функції $nsg(x_1)$ має вигляд $\mathbf{R}(\mathbf{S}^2(\boldsymbol{s},\boldsymbol{o}), \mathbf{S}^2(\boldsymbol{o}, \boldsymbol{I}_1^2))$.

***Приклад* 7.** Функція $x_1 \dot{-} x_2 = \begin{cases} x_1 - x_2, \text{ якщо } x_1 \geq x_2, \\ 0, \text{ якщо } x_1 \leq x_2, \end{cases}$ є ПРФ

Спочатку покажемо що функція $x_1 \dot{-} 1$ – ПРФ. Справді,

$0 \dot{-} 1 = 0 = \boldsymbol{o}(x_1)$;

$(x_1 + 1) \dot{-} 1 = x_1 = \boldsymbol{I}_1^2(x_1, x_2)$.

Операторний терм функції $x_1 \dot{-} 1$ має вигляд $\mathbf{R}(\boldsymbol{o}, \boldsymbol{I}_1^2)$.

Тепер покажемо що функція $f(x_1, x_2) = x_1 \dot{-} x_2$ – ПРФ. Маємо

$f(x_1, 0) = x_1 \dot{-} 0 = \boldsymbol{I}_1^1(x_1)$;

$f(x_1, x_2+1) = x_1 \dot{-} (x_2 + 1) = (x_1 \dot{-} x_2) \dot{-} 1 = f(x_1, x_2) \dot{-} 1$.

Отже, функція $f(x_1, x_2) = x_1 \dot{-} x_2$ виникає примітивною рекурсією із функцій $g(x_1) = \boldsymbol{I}_1^1(x_1)$ та $h(x_1, x_2, x_3) = x_3 \dot{-} 1$. Звідси ОТ функції $x_1 \dot{-} x_2$ має вигляд $\mathbf{R}(\boldsymbol{I}_1^1, \mathbf{S}^2(\mathbf{R}(\boldsymbol{o}, \boldsymbol{I}_1^2), \boldsymbol{I}_1^3))$.

***Приклад* 8**. Функція $f(x_1, x_2) = |x_1–x_2| = (x_1 \dot{-} x_2)+(x_2 \dot{-} x_1)$ – ПРФ.

***Приклад* 9**. Функція $f(x_1, x_2) = x_1–x_2 = \mu_{x_3}(|x_1–(x_2+x_3)| = 0)$ – ЧРФ.

***Приклад* 10.** Всюди невизначена функція $f_\varnothing$ – ЧРФ. ·

Справді, $f_\varnothing(x_1) = \mu_{x_1}(x_1+1 = 0)$, тому $f_\varnothing$ є значенням ОТ **M**(***s*).**

***Приклад* 11.** Функція $f(x_1, x_2) = [x_1/x_2]$ – ЧРФ. ·

Маємо $[x_1/x_2] = \mu_{x_3}(x_2 \cdot (x_3+1)>x_1) = \mu_{x_3}(nsg(x_2 \cdot (x_3+1) \dot{-} x_1) = 0)$.

Функція $[x_1/x_2]$ невизначена при $x_2 = 0$, тому вона не РФ, не ПРФ.

***Приклад* 12.** Функція $f(x_1) = [\sqrt{x_1}]$ – РФ.

Справді, $[\sqrt{x_1}]$ тотальна та $[\sqrt{x_1}] = \mu_{x_2}((x_2+1)\cdot(x_2+1)>x_1) = = \mu_{x_2}(nsg((x_2+1)\cdot(x_2+1) \dot{-} x_1) = 0)$.

Наведемо деякі елементарні властивості ПРФ і РФ. Для спрощення звичайно позначатимемо x_{n+1} та x_{n+2} як y та z відповідно.

Теорема 2.6.1. *Нехай* (n+1)*-арна функція g є ПРФ. Тоді ПРФ є* (n+1)*-арна функція f, задана умовою*

$$f(x_1,\ldots, x_n, y) = \sum_{k=0}^{y} g(x_1,\ldots,x_n,k).$$

Домовимося надалі вважати, що при $z<y$ $\sum_{k=y}^{z} a_k = 0$.

Теорема 2.6.2. *Нехай* (n+1)*-арна функція g є ПРФ. Тоді ПРФ є* (n+2)*-арна функція f, задана умовою*

$$f(x_1, \ldots, x_n, y, z) = \sum_{k=y}^{z} g(x_1,\ldots,x_n,k).$$

Теорема 2.6.3. *Нехай* (n+1)*-арна функція g та n-арні функції* α *і* β *є ПРФ. Тоді ПРФ є n-арна функція h, задана умовою*

$$h(x_1, \ldots, x_n) = \sum_{k=\alpha(x_1,\ldots,x_n)}^{\beta(x_1,\ldots,x_n)} g(x_1,\ldots,x_n,k).$$

Теореми 2.6.1–2.6.3 називають *теоремами підсумовування.* Замінивши в цих теоремах символ суми Σ на символ добутку Π, одержимо *теореми мультиплікації*.

Кажуть, що (n+1)-арна функція f отримується із (n+1)-арної функції g за допомогою операції *обмеженої мінімізації*, якщо вона задається так:

$$f(x_1, \ldots, x_n, y) = \begin{cases} \text{перше, починаючи з } 0, \text{ значення } z \text{ таке, що } z \le y \\ \text{та } g(x_1,\ldots,x_n) = 0, \text{ якщо таке значення } z \text{ існує,} \\ \text{значення } y, \text{ якщо такого значення } z \text{ не існує.} \end{cases}$$

Цей факт позначаємо так:

$f(x_1, \ldots, x_n, y) = \mu_{z \le y}((g(x_1, \ldots, x_n, z) = 0)$.

Теорема 2 (про обмежену мінімізацію). *Нехай* (n+1)*-арна функція g є ПРФ. Тоді ПРФ є* (n+1)*-арна функція f, задана умовою*

$f(x_1, \ldots, x_n, y) = \mu_{z \le y}((g(x_1, \ldots, x_n, z) = 0)$.

Наслідок. *Нехай функції* $g(x_1, \ldots, x_n, y)$ *та* $\alpha(x_1, \ldots, x_n)$ *є ПРФ. Тоді ПРФ є функція* $f(x_1, \ldots, x_n) = \mu_{y \le \alpha(x_1,\ldots,x_n)}((g(x_1, \ldots, x_n, y) = 0)$.

***Приклад* 13.** Довизначимо функцію $f(x_1, x_2) = [x_1/x_2]$ таким чином: $[x_1/0] = x_1$. Тоді $[x_1/x_2]$ є ПРФ.

Справді, значення $[a/b]$ рівне кількості нулів в послідовності $1 \cdot b \dot{-} a$, $2 \cdot b \dot{-} a$, ..., $a \cdot b \dot{-} a$. Маємо $[x_1/x_2] = \sum_{k=1}^{x_1} nsg(k \cdot x_2 \dot{-} x_1)$, тому $[x_1/x_2]$ є ПРФ за теоремою 2.6.3.

***Приклад* 14.** Функція $mod(x_1, x_2)$ – остача від ділення x_1 на x_2, є ЧРФ.

Справді, $mod(x_1, x_2) = x_1 \dot{-} (x_2 \cdot [x_1/x_2])$.

Беручи тут довизначену функцію $[x_1/x_2]$, дістанемо довизначену функцію $mod(x_1, x_2)$: $mod(x_1, 0) = 0$. Така довизначена функція $mod(x_1, x_2)$ є ПРФ.

***Приклад* 15.** Функція $f(x_1) = [\sqrt{x_1}]$ є ПРФ за теоремою 2.6.4.

Справді, $[\sqrt{x_1}] = \mu_{x_2 \le x_1}(nsg((x_2+1)\cdot(x_2+1) \dot{-} x_1) = 0)$.

***Приклад* 16.** Функція $nd(x)$ – кількість дільників числа x, – є ПРФ при довизначенні $nd(0) = 1$.

Справді, $nd(x) = \sum_{k=0}^{x} nsg(\mathrm{mod}(x,k))$.

***Приклад* 17.** Функція $\pi(x)$ – кількість простих чисел, які не більші за число x, – є ПРФ.

Справді, $\pi(x) = \sum_{k=0}^{x} nsg(|nd(x) - 2|)$.

***Приклад* 18.** Функція $p(x)$ – x-ве просте число, – є ПРФ

За визначенням $p(0) = 2$, $p(1) = 3$, $p(2) = 5$ і т.д.

В загальному випадку $p(x) = \mu_{y \le 2^{2^x}}(|\pi(y) - (x+1)| = 0)$.

Доведення того, що $p(x) \le 2^{2^x}$, проведемо індукцією по x. Для $x = 0$ маємо $p(0) = 2 \le 2^{2^0}$. Нехай $p(x) \le 2^{2^x}$ для всіх $x \le k$. Покажемо $p(k+1) \le 2^{2^{k+1}}$. Розглянемо число $b = p(0)\cdot p(1)\cdot \ldots \cdot p(k)+1$. За припущенням індукції $p(0)$

$\le 2^{2^0}$, …, $p(k) \le 2^{2^k}$. Тому дістаємо $b \le 2^{2^0+2^1+...+2^k} + 1 < 2^{2^{k+1}}$. Але кожний простий дільник числа b більший за $p(k)$, тому $p(k+1) \le b \le 2^{2^{k+1}}$.

***Приклад* 19.** Функція $ex(x, y)$ – степінь числа $p(x)$ в числі y, – є ПРФ при довизначенні $ex(x, 0) = 0$. Справді, $ex(x, y) = \mu_{z \le y}(nsg(\mathrm{mod}(y, p(x)^{z+1})) \cdot sg(y) = 0)$.

ВПРАВИ

1. Чи може бути тотальною функція $\mathbf{S}^{n+1}(g, g_1, ..., g_n)$, якщо g нетотальна ?
2. Чи може бути тотальною функція $\mathbf{S}^{n+1}(g, g_1, ..., g_n)$, якщо одна з функцій $g_1, ..., g_n$ нетотальна ?
3. Чи може бути тотальною функція $\mathbf{R}(g, h)$, якщо g нетотальна ?
4. Чи може бути тотальною функція $\mathbf{R}(g, h)$, якщо h нетотальна ?
5. Чи може бути тотальною функція $\mathbf{M}(g)$, якщо g нетотальна ?
6. Промоделюйте операції $\mathbf{S}^{n+1}$, **R**, **M** за допомогою систем Поста.
7. Доведіть, що функція $НСК(x_1, x_2)$ (найменше спільне кратне чисел x_1 та x_2) є ПРФ при довизначенні $НСК(n, 0) = НСК(0, n) = 0$.
8. Доведіть, що функція $НСД(x_1, x_2)$ (найбільший спільний дільник чисел x_1 та x_2) є ПРФ при довизначенні $НСД(n, 0) = НСД(0, n) = 0$.
9. Доведіть, що при належному довизначенні наступні функції є ПРФ:

1) $\sigma(x)$ – сума дільників числа x;

2) $spd(x$сума простих дільників числа x;

3) $kpd(x)$ – кількість простих дільників числа x.

10. Вкажіть ОТ алгебри n-арних ЧРФ для функцій:

1) $f(x_1) = (x_1)!$; 2) $f(x_1, x_2) = (x_2 + 2)!$;

3) $f(x_1, x_2, x_3) = (x_2 + x_3)!$; 4) $f(x_1) = (2x_1)!!$;

5) $f(x_1, x_2) = (x_1 + 1)^{x_2}$; 6) $f(x_1, x_2) = x_1^{x_2+3}$;

7) $f(x_1, x_2, x_3) = x_2^{x_1+x_3}$; 8) $f(x_1) = [\log_2(x_1)]$;

9) $f(x_1, x_2) = [\log_3(x_1+x_2)]$; 10) $f(x_1) = [\sqrt[3]{x_1}]$;

11) $f(x_1) = [x_1\sqrt{3}]$; 12) $f(x_1, x_2) = [\log_{x_1}(x_2)]$.

3.4. ПРАКТИЧНА РОБОТА 11. ЕКВІВАЛЕНТНІ ПЕРЕТВОРЕННЯ ФОРМУЛ. РЕЗОЛЮЦІЙНЕ ЧИСЛЕННЯ ЛОГІКИ ПЕРШОГО ПОРЯДКУ

ВИКОРИСТАНО МАТЕРІАЛ:

1. ***Зубенко В. В. Алгоритмічні процедури: навчальний посібник / В. В. Зубенко, О. В. Шишацька, Ф. А. Асроров. – Київ, 2023. 56 с.***
2. ***С.С.Шкільняк С.С. Теорія алгоритмів. Приклади та задачі: Навчальний посібник. – Київ: Видавничо-поліграфічний центр "Київський університет", 2012. – 144 с. (сторінка 42-50)***

Кодування та нумерації. Канторові нумерації

Кодування множини A на множині B – це ін'єктивне та сюр'єктивне відображення $\varphi : A \to B$ таке, що існують алгоритми $\aleph$ та $\Re$:

– для кожного $a \in A$ маємо $\aleph(a) \in \varphi(a)$;

– для кожного $b \in B$ маємо $\Re(b) = \varphi^{-1}(b)$.

Нумерацією множини *A* називають сюр'єктивне функціональне відображення $\xi: N \to A$.

Однозначною нумерацією множини *A* називають бієктивне відображення $\xi : N \to A$.

Нумерація $\xi : N \to A$ *ефективна*, якщо існують алгоритми $\aleph$ та $\Re$:

– для кожного $a \in A$ маємо $\aleph(a) \in \xi^{-1}(a)$;

– для кожного $n \in N$ маємо $\Re(n) = \xi(n)$.

Таким чином, $\xi : N \to A$ – ефективна нумерація $\Leftrightarrow \xi^{-1} : A \to N$ – кодування *A* на *N*.

Введемо однозначні ефективні нумерації пар та *n*-ок натуральних чисел, які називаються *канторовими нумераціями*.

Всі пари натуральних чисел розташуємо в послідовність так:

пара (x, y) *передує парі* $(u, v) \Leftrightarrow x+y<u+v$ або ($x+y = u+v$ та $x<u$).

Номер пари (x, y) в такій послідовності позначають $C(x, y)$ та називають *канторовим номером* пари (x, y).

Зрозуміло, що $C(x, y) = [(x+y+1)\cdot(x+y)/2]+x$

Ліву та праву компоненти пари з номером *n* позначають відповідно $\boldsymbol{l}(n)$ та $\boldsymbol{r}(n)$. Функції $\boldsymbol{l}(n)$ та $\boldsymbol{r}(n)$ називають відповідно лівою та правою координатними функціями.

Функція $C(x, y)$ задає бієкцію $N\times N \to N$, пара функцій $(\boldsymbol{l}(n), \boldsymbol{r}(n))$ задає бієкцію $N \to N\times N$.

Функції *C*, $\boldsymbol{l}$ та $\boldsymbol{r}$ зв'язані тотожностями:

$C(\boldsymbol{l}(n), \boldsymbol{r}(n)) = n$;

$\boldsymbol{l}(C(x, y)) = x$;

$\boldsymbol{r}(C(x, y)) = y$.

Маючи нумерацію пар натуральних чисел, можна ввести нумерацію *n*-ок натуральних чисел для довільного $n>2$:

$C^n(x_1, x_2,..., x_n) = C^{n-1}(C(x_1, x_2), x_3,..., x_n)$

Зрозуміло, що $C^n(x_1, x_2,..., x_n) = C(...C(C(x_1, x_2), x_3),...), x_n)$.

Координатні функції C_{n1} , ..., C_{nn} вводимо так:

Нехай $C^n(x_1, x_2,..., x_n) = m$; тоді

$C_{n1}(m) = x_1$; $C_{n2}(m) = x_2$; ..., $C_{nn}(m) = x_n$.

Для функцій C^n, C_{n1} , ..., C_{nn} справджуються такі тотожності:

$C^n(C_{n1}(x), ..., C_{nn}(x)) = x$;

$C_{nk}(C^n(x_1, x_2,..., x_n)) = x_k$, $1\le k\le n$.

Теорема 1. 1) *Функції* $C(x, y)$, $\boldsymbol{l}(n)$ *та* $\boldsymbol{r}(n)$ *є ПРФ.*

2) *Функції* C^n, C_{n1} , ..., C_{nn} *є ПРФ.*

Приклад* 1.** Знайдемо ***l(100) та ***r***(100). Для x = ***l***(100) та y = ***r***(100) маємо рівняння $C(x, y) = 100$. Нехай $x+y = p$. Тоді p є найбільшим натуральним числом, для якого $p\cdot(p+1) \le 2\cdot 100$. Звідси $p = 13$. Маємо $x+y = 13$, $x = 100 - [(13\cdot 14)/2] = 9$, $y = 13-9 = 4$. Отримуємо ***l***(100) = 9 та ***r***(100) = 4.

***Приклад* 2.** Розв'яжемо рівняння $C^4(x, y, z, v) = 207$. За визначенням $C(C(C(x, y), z), v) = 207$. Нехай $C(C(x, y), z) = a$, тоді $C(a, v) = 207$. Нехай $a+v = p$. Тоді p є найбільшим натуральним числом, для якого $p\cdot(p+1)\le 2\cdot 207$. Звідси $p = 19$, звідки $a = 17$ та $v = 2$. Маємо $C(C(x, y), z) = 17$. Нехай $C(x, y) = b$, тоді $C(b, z) = 17$. Нехай $b+z = q$. Тоді q є найбільшим натуральним числом, для якого $q\cdot(q+1) \le 2\cdot 17$. Звідси $q = 5$, звідки $b = 2$ та $z = 3$. Маємо $C(x, y) = 2$, звідки $x = 1$ та $y = 0$.

***Приклад* 3.** Задамо однозначну ефективну нумерацію всіх скінченних послідовностей натуральних чисел на основі такого кодування $\kappa: \bigcup_{k\ge 0} N^k \to N$:

$$\kappa(\varnothing) = 0;\ \kappa(a_1, ..., a_n) = C(C^n(a_1, ..., a_n), n-1)+1.$$

Зрозуміло, що таке відображення бієктивне. Тепер обернене відображення $\eta = \kappa^{-1}$ – шукана однозначна нумерація.

***Приклад* 4.** Однозначну ефективну нумерацію всіх скінченних послідовностей натуральних чисел можна задати на основі кодування $\sigma: \bigcup_{k\ge 0} N^k \to N$:

$$\sigma(\varnothing) = 0;\ \sigma(a_1, ..., a_n) = 2^{a_1} + 2^{a_1+a_2+1} + ... + 2^{a_1+a_2+...+a_n+n-1}.$$

Бієктивність відображення σ випливає із однозначності подання кожного натурального числа в двійковій системі числення. Обернене відображення $\lambda = \sigma^{-1}$ – шукана однозначна нумерація.

***Приклад* 5**. Однозначну ефективну нумерацію всіх непорожніх скінченних послідовностей натуральних чисел задамо на основі кодування $\nu: \bigcup_{k\ge 1} N^k \to N$, яке є модифікацією кодування σ:

$$\nu(a_1, ..., a_n) = 2^{a_1} + 2^{a_1+a_2+1} + ... + 2^{a_1+a_2+...+a_n+n-1} - 1.$$

Обернене відображення $\zeta = \nu^{-1}$ – шукана однозначна нумерація.

***Приклад* 6**. Ефективну нумерацію ϕ множини формул довільної мови 1-го порядку із зліченним алфавітом [14] введемо наступним чином. Занумеруємо множини предметних імен (змінних) x_0, x_1, ..., константних символів c_0, c_1, ... , функціональних символів f_0, f_1, ... та предикатних символів p_0, p_1,... .

Кодування τ термів та формул ξ задамо так:

$\tau(x_k) = 3\cdot k$;

$\tau(c_k) = 3\cdot k+1$;

$\tau(f_k t_1...t_n) = 3\cdot C(C^{n+1}(k, \tau(t_1),..., \tau(t_n)), n-1)+2$;

$\xi(p_k t_1...t_n) = 4\cdot C(C^{n+1}(k, \tau(t_1),..., \tau(t_n)), n-1)$;

$\xi(\neg\Phi) = 4\cdot\xi(\Phi)+1$;

$\xi(\Phi\vee\Psi) = 4\cdot C(\xi(\Phi),\xi(\Psi))+2$;

$\xi(\exists x_k\Phi) = 4\cdot C(k, \xi(\Phi))+3$.

Номером (*індексом*) довільної формули Φ вважатимемо її код $\xi(\Phi)$. Всі ті натуральні числа, які не є кодами формул, вважатимемо номером формули $x_0 = x_0$. Зрозуміло, що так введена нумерація $\phi = \xi^{-1}$ неоднозначна. Формулу з номером n будемо позначати ϕ_n .

Кодувати довільну скінченну послідовність натуральних чисел. дозволяє також *функція Гьоделя*

$\Gamma(x, y) = mod(\boldsymbol{l}(x), 1+(y+1)\cdot\boldsymbol{r}(x))$.

З визначення випливає, що $\Gamma(x, y)$ є ПРФ.

Теорема 3.1.2 (про основну властивість функції Гьоделя). *Для довільної скінченної послідовності натуральних чисел* $b_0, b_1, .., b_n$ *існує натуральне число* t *таке, що* $\Gamma(t, i) = b_i$ *для всіх* $i \in \{0,..., n\}$.

Використання функції Гьоделя дає змогу промоделювати операцію примітивної рекурсії операціями суперпозиції та мінімізації:

Теорема 3.1.3. *Функція* $f = \mathbf{R}(g, h)$ *може бути отримана із функцій* g, h, *базових функцій і функцій* $+, \times$ $ma \dot{-}$ *за допомогою скінченної кількості застосувань операцій* $\mathbf{S}^{n+1}$ *та* $\mathbf{M}$.

Наслідок. *Клас ЧРФ збігається з класом функцій, отриманих із функцій* $\boldsymbol{o}$, $\boldsymbol{s}$, $\boldsymbol{I}_m^n$, $+, \times$ $ma \dot{-}$ *за допомогою операцій* $\mathbf{S}^{n+1}$ *та* $\mathbf{M}$.

ВПРАВИ

1. Знайдіть $\boldsymbol{l}(150)$, $\boldsymbol{r}(150)$ та $\boldsymbol{l}(200)$, $\boldsymbol{r}(200)$.

2. Розв'яжіть рівняння $C(x, y) = 2012$.

3. Розв'яжіть наступні рівняння:

1) $C^3(x,y,z) = 131$; 2) $C^3(x,y,z) = 133$;

3) $C^3(x,y,z) = 226$; 4) $C^3(x,y,z) = 212$;

5) $C^4(x,y,z,v) = 123$; 6) $C^4(x,y,z,v) = 333$;

7) $C^4(x,y,z,v) = 228$; 8) $C^4(x,y,z,v) = 282$.

4. Функція $f : N \to N$ називається функцією великого розмаху (ФВР), якщо для кожного $a \in N$ рівняння $f(x) = a$ має нескінченну кількість розв'язків. Доведіть, що:

1) функції $\boldsymbol{l}(x)$ та $\boldsymbol{r}(x)$ є ФВР;

2) для кожної рекурсивної ФВР $L(x)$ існує рекурсивна ФВР $R(x)$ така, що пара (L, R) задає бієкцію $N \to N \times N$;

3) для $L(x)$, $R(x)$ та функції $C(x, y) = \mu_t(x = L(t)\ \&\ y = R(t))$ справджуються тотожності $C(L(n), R(n)) = n$; $L(C(x, y)) = x$; $R(C(x, y)) = y$. Це означає, що функції L, R та C грають роль канторових нумераційних функцій $\boldsymbol{l}$, $\boldsymbol{r}$ та C.

5. Наведіть конкретні приклади ефективних нумерацій множини формул мови арифметики, множини формул мови теорії множин та множини формул мови частково впорядкованих множин.

6. Сформулюйте та доведіть теорему про моделювання операції примітивної рекурсії для випадку фінарних ЧРФ.

Ефективні нумерації формальних моделей алгоритмів та АОФ

Розглянемо приклади ефективних нумерацій формальних моделей алгоритмів та алгоритмічно обчислюваних функцій.

***Приклад* 1.** Однозначну ефективну нумерацію всіх МНР-програм задамо на основі кодування МНР-програм як скінченних послідовностей команд МНР. Кодування команд θ задамо так:

$\theta(Z(n)) = 4\cdot n$;

$\theta(S(n)) = 4\cdot n+1$;

$\theta(T(m, n)) = 4\cdot C(m, n)+2$;

$\theta(J(m, n, q+1)) = 4\cdot C(C(m, n), q)+3$.

Таке θ – бієктивне відображення множини всіх команд МНР на множину N.

Використовуючи розглянуту вище бієкцію $\nu : \bigcup_{k\geq 1} N^k \to N$, задамо кодування τ всіх МНР-програм так. Якщо P – МНР-програма $I_1 I_2 ... I_k$, то $\tau(P) = \nu(\theta(I_1),..., \theta(I_k))$. Відображення ν та θ бієктивні, тому τ теж бієкція. Тоді $\varphi = \tau^{-1}$ – шукана однозначна нумерація.

Нумерація φ ефективна. Справді, за кожною МНР-програмою P ефективно обчислюється її номер $\tau(P)$. З іншого боку, за кожним $n\in N$ ефективно визначається МНР-програма $P = \varphi(n)$. Наведемо відповідний алгоритм. Спочатку подамо число n+1 як суму зростаючих степенів числа 2: $n+1 = 2^{b_1} + 2^{b_2} +...+ 2^{b_k}$, де $0 \leq b_1 <...< b_k$. Далі визначимо таку послідовність чисел $a_1, ..., a_k$: $a_1 = b_1$, $a_{i+1} = b_{i+1} - b_i - 1$ для $1 < i < k$. За числами $a_1, ..., a_k$ як за кодами команд МНР відновимо відповідні команди. Послідовність цих команд і є шукана $P = \varphi(n)$.

МНР-програму з кодом n позначатимемо P_n.

Приклад 1.1. Знайдемо код МНР-програми P, яка обчислює бінарну функцію $x+y$. Використовуємо приклад 3 розділу 2.1:

1) $J(1, 2, 5)$;

$\theta(J(1, 2, 5)) = 4\cdot C(C(1,2),4)+3 = 4\cdot C(7,4)+3 = 4\cdot 73+3 = 295$;

2) $S(0)$;

$\theta(S(0)) = 4\cdot 0+1 = 1$;

3) $S(2)$;

$\theta(S(2)) = 4\cdot 2+1 = 9$;

4) $J(0, 0, 1)$;

$\theta(J(0, 0, 1)) = 4\cdot C(C(0,0),0)+3 = 4\cdot C(0,0)+3 = 4\cdot 0+3 = 3$.

Тепер $\tau(P) = \nu(295, 1, 9, 3) = 2^{295} + 2^{297} + 2^{307} + 2^{311} - 1$.

Приклад 1.2. Знайдемо $P_{5119} = \varphi(5119)$. Маємо 5119+1 = 5120 = = $2^{10}+2^{12}$, звідки $a_1 = 10$, $a_2 = 12-10-1 = 1$. Але $10 = 2+4\cdot C(1,0)$, тому МНР-програма P_{5119} така:

1) $T(1, 0)$;

2) $S(0)$.

***Приклад* 2.** Ефективну нумерацію всіх МТ задамо на основі кодування МТ. Кожну МТ можна задати послідовністю її команд такою, що 1-а команда містить в лівій частині символ q_0, остання команда містить в правій частині символ q^*. Така нумерація МТ неоднозначна, бо множину команд МТ можна впорядкувати у вигляді послідовності з зазначеною властивістю багатьма способами.

Спочатку занумеруємо внутрішні стани МТ і символи стрічки. Нехай $Q = \{q_0,..., q_t\}$, $T = \{a_0,..., a_m\}$. Задамо кодування μ команд МТ:

$\mu(q_i a_j \to q_k a_l) = 3 \cdot C^4(i, j, k, l)$;

$\mu(q_i a_j \to q_k a_l L) = 3 \cdot C^4(i, j, k, l)+1$;

$\mu(q_i a_j \to q_k a_l R) = 3 \cdot C^4(i, j, k, l)+2$.

Таке μ є бієктивним відображенням множини всіх можливих команд машин Тьюрінга на N. Використовуючи розглянуту вище бієкцію $\nu: \bigcup_{k\geq 1} N^k \to N$, визначимо код МТ M, заданої послідовністю команд $I_1 I_2 ... I_k$, таким чином: $\rho(M) = \nu(\mu(I_1),..., \mu(I_k))$. Але ν та μ бієктивні, тому ρ теж бієкція. Тоді $\varphi = \rho^{-1}$ – шукана однозначна нумерація послідовностей команд МТ, але неоднозначна нумерація всіх МТ. Неважко переконатись, що така нумерація ефективна.

Приклад 2.1. Знайдемо код МТ M, яка обчислює функцію $x+y$. Нехай $a_0 = \lambda$, $a_1 = |$, $a_2 = \#$, $q^* = q_2$.

Використаємо приклад 1 розділу 2.2:

$q_0| \to q_0|R$;

$\mu(q_0| \to q_0|R) = 3 \cdot C^4(0,1,0,1)+2 = 3 \cdot C(2,1)+2 = 3 \cdot 8+2 = 26$;

$q_0\# \to q_0|R$;

$\mu(q_0\# \to q_0|R) = 3 \cdot C^4(0,2,0,1)+2 = 3 \cdot C(9,1)+2 = 3 \cdot 64+2 = 194$;

$q_0\lambda \to q_1\lambda L$;

$\mu(q_0\lambda \to q_1\lambda L) = 3 \cdot C^4(0,0,1,0)+1 = 3 \cdot C(1,0)+1 = 3 \cdot 2+1 = 7$;

$q_1| \to q^*\lambda$;

$\mu(q_1| \to q^*\lambda) = 3 \cdot C^4(1,1,2,0) = 3 \cdot C(25,0) = 3 \cdot 350 = 1050$.

Тепер $\rho(M) = \nu(26, 194, 7, 1050) = 2^{26} + 2^{221} + 2^{229} + 2^{1280} - 1$.

***Приклад* 3.** Ефективну нумерацію ϕ m-арних ЧРФ для всіх $m \geq 1$ задамо на основі кодування γ ОТ алгебри ЧРФ. Кожну ЧРФ можна описати нескінченною кількістю різних ОТ, тому ϕ неоднозначна.

Кодування γ операторних термів задамо так:

$\gamma(\mathbf{o}) = 0$;

$\gamma(\mathbf{s}) = 4$;

$\gamma(\mathbf{I}_m^n) = 4 \cdot C(n-m, m-1) + 8$;

$\gamma(\mathbf{S}^{n+1}(t_0, t_1,..., t_n)) = 4 \cdot C(C^{n+1}(\gamma(t_0), \gamma(t_1),..., \gamma(t_n)), n-1) +1$;

$\gamma(\mathbf{R}(t_0, t_1)) = 4 \cdot C(\gamma(t_0), \gamma(t_1)) +2$;

$\gamma(\mathbf{M}(t)) = 4 \cdot \gamma(t) +3$.

Число $n \in N$ вважаємо *номером* ЧРФ f, якщо n є кодом якогось ОТ, і значенням цього ОТ є функція f. Числа, які не є кодами ОТ, та коди тих ОТ, які не задають ЧРФ, вважаємо номерами всюди невизначеної функції $f_\varnothing$. Отже, за кожною ЧРФ ефективно знаходиться її номер. З іншого боку, за кожним $n \in N$ ефективно визначається ЧРФ f така, що $\phi(n) = f$: за n як за кодом будуємо ОТ; якщо ОТ з таким кодом існує та задає ЧРФ, то $\phi(n)$ – ця ЧРФ f; якщо ОТ з таким кодом існує, але не задає ЧРФ, або якщо ОТ з таким кодом не існує, то $\phi(n) = f_\varnothing$. Тому ϕ – ефективна нумерація.

Приклад 3.1. Знайдемо код ОТ алгебри *n*-арних ЧРФ для всюди невизначеної функції $f_\varnothing$. Згадуючи приклад 9 розділу 2.6, для $f_\varnothing$ маємо ОТ $\mathbf{M}(\boldsymbol{s})$, звідки $\gamma(\mathbf{M}(\boldsymbol{s})) = 4\cdot\gamma(\boldsymbol{s})+3 = 4\cdot 4+3 = 19$.

Приклад 4. Ефективну неоднозначну нумерацію всіх ПРФ задамо на основі наступного кодування π для ОТ алгебри ПРФ:

$\pi(\boldsymbol{o}) = 0$;

$\pi(\boldsymbol{s}) = 3$;

$\pi(\boldsymbol{I}^{n}_{m}) = 3\cdot C(n-m, m-1) + 6$;

$\pi(\mathbf{S}^{n+1}(t_0, t_1,..., t_n)) = 3\cdot C(C^{n+1}(\pi(t_0), \pi(t_1),..., \pi(t_n)), n-1)+1$;

$\pi(\mathbf{R}(t_0, t_1)) = 3\cdot C(\pi(t_0), \pi(t_1))+2$.

Число $n\in N$ вважаємо *номером* ПРФ *f*, якщо *n* є кодом якогось ОТ та значенням цього ОТ є функція *f*. Числа, які не є кодами ОТ, та коди тих ОТ, які не задають ПРФ, вважаємо номерами функції $\boldsymbol{o}$.

Приклад 4.1. Знайдемо код ОТ алгебри ПРФ для $f(x_1, x_2) = x_1+x_2$. Згідно прикладу 2 розділу 2.6, для x_1+x_2 маємо ОТ $\mathbf{R}(\boldsymbol{I}^{1}_{1}, \mathbf{S}^2(\boldsymbol{s}, \boldsymbol{I}^{3}_{3}))$, звідки

$\pi(\mathbf{R}(\boldsymbol{I}^{1}_{1}, \mathbf{S}^2(\boldsymbol{s}, \boldsymbol{I}^{3}_{3}))) = 3\cdot C(\pi(\boldsymbol{I}^{1}_{1})$,

$\pi(\mathbf{S}^2(\boldsymbol{s}, \boldsymbol{I}^{3}_{3}))) +2 = 3\cdot C(6, 3\cdot C(C(\pi(\boldsymbol{s}), \pi(\boldsymbol{I}^{3}_{3})), 0)+1)+2 =$

$= 3\cdot C(6, 3\cdot C(C(3, 15), 0)+1)+2 = 3\cdot C(6, 3\cdot C(174, 0)+1)+2 =$

$= 3\cdot C(6, 3\cdot 15399+1)+2 = 3\cdot C(6, 46198)+2 = 3\cdot 1067427916+2 =$

$= 3202283748+2 = 3202283750$.

Приклад 5. Ефективну нумерацію всіх програмованих на *N n*-арних функцій задамо на основі кодування операторних термів ППА-AR-*N*. Така нумерація неоднозначна. Єдина істотна відмінність від нумерації прикладу 3 – інше кодування γ операторних термів ППА-AR-*N* :

$\gamma(\boldsymbol{o}) = 0$;

$\gamma(\boldsymbol{s}) = 4$;

$\gamma(+) = 8$;

$\gamma(\cdot) = 12$;

$\gamma(\dot{-}) = 16$:

$\gamma(\boldsymbol{I}^{n}_{m}) = 4\cdot(C(n-m, m-1) +5)$;

$\gamma(\mathbf{S}^{n+1}(t_0, t_1,..., t_n)) = 4\cdot C(C^{n+1}(\gamma(t_0), \gamma(t_1),..., \gamma(t_n)), n-1)+1$;

$\gamma({}^{N}\Delta(t_0, t_1, t_2)) = 4\cdot C^3(\gamma(t_0), \gamma(t_1), \gamma(t_2))+2$;

$\gamma({}^{N}☼(t_0, t_1)) = 4\cdot C(\gamma(t_0), \gamma(t_1))+3$.

Приклад 6. Ефективну нумерацію всіх МНР-обчислюваних функцій фіксованої арності *n* задамо на основі кодування МНР-програм (приклад 1). Номером *n*-арної МНР-обчислюваної функції *f* буде код МНР-програми, яка обчислює функцію *f*. Кожна *n*-арна МНР-обчислювана функція може задаватися нескінченною кількістю різних МНР-програм, тому така нумерація неоднозначна.

Приклад 7. Ефективну нумерацію всіх МНР-обчислюваних функцій можна ввести на основі кодування МНР-програм. Номером *n*-арної функції *f* буде число $C(k, n)$, де *k* – код МНР-програми для *f*.

***Приклад* 8.** Ефективну нумерацію всіх МТ-обчислюваних функцій фіксованої арності *n* задамо на основі кодування МТ. Номером *n*-арної МТ-обчислюваної функції *f* буде код МТ, яка обчислює *f*. Кожна *n*-арна МТ-обчислювана функція може задаватися нескінченною кількістю різних МТ, тому така нумерація неоднозначна.

***Приклад* 9.** Ефективну нумерацію всіх обчислюваних за Тьюрінгом функцій введемо на основі кодування МТ. *Номером* МТ-обчислюваної функції *f* арності *n* буде число $C(k, n)$, де k – код МТ для f.

Вправи

1. Вкажіть МНР-програму та її код для функцій:

1) $f(x) = 2x$; 2) $f(x,y) = x–y$;

3) $f(x) = x/2$; 4) $f(x) = [x/2]$;

5) $f(x) = \mathrm{nsg}(x)$; 6) $f(x) = \mathrm{sg}(x)$;

7) $f(x, y) = \mathrm{sg}(x+y)$; 8) $f(x, y) = \mathrm{nsg}(x+y)$;

9) $f(x, y) = \mathrm{sg}(x\cdot y)$; 10) $f(x, y) = \mathrm{nsg}(x\cdot y)$;

11) $f(x, y) = \min(x, y)$; 12) $f(x, y) = \max(x, y)$.

2. Вкажіть МНР-програму та її код для предикатів:

1) "x – непарне число"; 2) "x – парне число";

3) "$x = 3$"; 4) "$x \neq y$".

3. Визначіть всі МНР-програми з кодами від 0 до 100.

4. Вкажіть МТ та її код для функцій та предикатів:

1) $f(x) = \mathrm{sg}(x)$; 2) $f(x) = \mathrm{nsg}(x)$;

3) $f(x, y) = \mathrm{sg}(x+y)$; 4) $f(x, y) = \mathrm{nsg}(x+y)$;

5) $f(x) = \mathrm{sg}[x/2]$; 6) $f(x) = \mathrm{nsg}(x/2)$;

7) "x – непарне число"; 8) "x – парне число".