МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Закарпатський державний університет
Факультет інформатики

Кафедра інформаційних управляючих систем та технологій

Методичний посібник до курсу

“ Теорія алгоритмів та математичні основи представлення знань”

зі спеціальності
“інформаційні управляючі системи та технології”

частина II

Ужгород-2005

Методичний посібник до курсу “Теорія алгоритмів та математичні основи представлення знань” для студентів ІІІ курсу стаціонарної та заочної форми навчання містить деякі теоретичні відомості, завдання для самостійної та індивідуальної роботи студентів, літературу.

Методичний посібник розроблено і складено асистентом кафедри інформаційних управляючих систем та технологій **Копча-Горячкіною Г.Е.**

Відповідальний за випуск доктор технічних наук, професор **Василенко Ю.А.**

Друкується за рішенням кафедри інформаційних управляючих систем та технологій від 29.12.05р., протокол № 4.

Вступ

Існують три основні типи універсальних алгоритмічних моделей, які є еквівалентними між собою. Кожна з цих алгоритмічних моделей являється уточненням інтуїтивного поняття алгоритму.

Перший тип алгоритмічних моделей – це перетворення слів у довільних алфавітах, в яких елементарними операціями є підстановки, тобто заміни частини слова іншим словом. Переваги цього типу – в його максимальній абстрактності та можливості застосовувати поняття алгоритму до об'єктів довільної (не обов'язково числової) природи. Приклади моделей цього типу – нормальні алгоритми Маркова.

Другий тип ґрунтується на уявленні про алгоритм як про деякий детермінований пристрій, здатний виконувати в кожен окремий момент лише дуже примітивні операції (тобто кроки є елементарними, а алгоритм однозначним). Основною теоретичною моделлю цього типу є машина Тюрінга.

Третій тип зв'язує поняття алгоритму з найтрадиційнішими поняттями математики – обчисленнями та числовими функціями. Найбільш розвинена й вивчена модель цього типу – рекурсивні функції.

Є ще четвертий тип універсальної алгоритмічної моделі – алгоритмічна система Кліні, яка є похідною від рекурсивних функцій.

Алгоритмічна система Маркова

В основі цієї алгоритмічної системи лежить поняття нормального алгоритму, введеного Марковим. Це поняття алгоритму пов'язане з асоціативним численням, що будується на множині всіх слів у заданому алфавіті.

Алфавітом називається будь-яка скінчена послідовність символів, що називаються літерами.

Словом у даному алфавіті називається довільна скінчена послідовність букв з даного алфавіту.

Порожнє слово позначається λ.

Якщо слово α є частиною слова β, то кажуть, що α входить в слово β, або, що α є *підсловом* слова β.

Операції асоціативного числення визначаються як система підстановок, за допомогою яких одні слова перетворюються на інші.

Нехай дано деяке слово γ в алфавіті Σ і нехай α є підсловом слова γ. Підстановка $\alpha \to \beta$ означає заміну підслова α в слові γ на деяке слово β.

Асоціативним численням називається множина слів в деякому алфавіті із деякою скінченою системою підстановок.

Приклад 1. Алфавіт $\{a,b,c,d,e\}$ та система підстановок

$$ac \to ca$$

$ad \to da$
$bc \to cb$
$bd \to db$
$abac \to abacc$ задають асоціативне числення.

Два слова α і β називаються *еквівалентними*, якщо одне з них можна утворити з іншого послідовним застосуванням деяких підстановок. Позначають $\alpha \sim \beta$.

Якщо α є частиною слова γ і $\alpha \sim \beta$, то після підстановки $\alpha \to \beta$ одержимо слово, еквівалентне до γ.

У кожному асоціативному численні існує нескінченна множина різних слів, яку за допомогою еквівалентності можна розбити на підмножини, які не перетинаються.

Дедуктивним ланцюжком називається послідовність слів $\alpha_1, \alpha_2, \ldots, \alpha_n$, в якій кожне наступне слово одержується з попереднього за допомогою одноразового застосування допустимої підстановки.

Два сусідні слова в дедуктивному ланцюжку називаються *суміжними*. Підстановка $\alpha \to \beta$ називається *допустимою для слова* γ, якщо α є підсловом слова γ.

У кожному асоціативному численні виникає своя проблема слів, тобто для будь-яких двох слів треба визначити еквівалентні вони чи ні (чи існує дедуктивний ланцюжок від одного слова до іншого).

Оскільки в кожному асоціативному численні існує нескінченна множина різних слів, то питання еквівалентності потребує перебору нескінченної множини варіантів.

За допомогою алгоритму перебору можна вирішити *обмежену проблему слів*, тобто встановити, чи можна одне слово одержати з іншого застосуванням допустимих підстановок не більше k разів, де k–наперед задане фіксоване число. Для цього досить побудувати всі слова, суміжні з одним із заданих слів; потім для кожного з утворених слів побудувати слова, суміжні з ним, і т.д. k разів. В результаті одержимо список всіх слів, які можуть бути утворені із заданого слова за допомогою допустимих підстановок, застосованих не більше k разів. Якщо друге задане слово буде в цьому списку, то відповідь позитивна, якщо ні – негативна.

Для *необмеженої проблеми* слів цей спосіб непридатний, оскільки довжина дедуктивного ланцюжка, якщо він існує, може бути як завгодно великою і тому не можна зазначити таке скінчене число k, що гарантовано вирішує цю проблему шляхом перебору, тобто взагалі невідомо чи існує таке скінчене k. Тому для вирішення необмеженої проблеми слів потрібно застосовувати ідеї, які ґрунтуються на аналізі механізму перетворення слів з використанням допустимих підстановок.

Наприклад, іноді можна виявити властивості, які є незмінними для всіх слів дедуктивного ланцюжка.

Властивості, що залишаються незмінними для всіх слів дедуктивного ланцюжка, називаються *дедуктивно-інваріантними*.

У прикладі 1 у кожній з допустимих підстановок числення ліва і права частини містять ту саму кількість підслів a.

Отже, у будь-якому дедуктивному ланцюжку всі слова також мають містити ту саму кількість підслів a. На основі цього дедуктивного інваріанта можна встановити, які слова не можуть бути еквівалентними.

Наприклад, слова *abacuac* і *abaaaac* не еквівалентні в заданому численні.

До проблеми слів зводяться багато геометричних, алгебраїчних та логічних задач.

Наприклад, будь-яка формула логіки висловлювань або логіки предикатів може розглядатися як слово в деякому алфавіті, що містить логічні символи, висловлювання, предикати і предметні змінні. А процес еквівалентних перетворень або виведення логічного висновку може трактуватись як перетворення слів, а роль допустимих підстановок будуть відігравати логічні закони або аксіоми. Тобто питання про виводимість якої-небудь формули є питанням існування дедуктивних ланцюжків, які ведуть від слів, що є посилками, до слів, які є наслідками.

Кажуть, що *задано алгоритм в алфавіті* Σ, який є *застосовним* до слова α і перетворює його на β, якщо, виходячи із слова α і діючи відповідно до розпоряджень, цей алгоритм дає β.

Множина слів, до яких є застосовним деякий алгоритм, називається *областю його застосування*.

Два алгоритми називаються *еквівалентними* в деякому алфавіті, якщо області їх застосування та результат перетворення ними будь-якого слова зі спільної області збігаються.

*Означення I нормального алгоритму***.** Нехай задано алфавіт Σ і впорядковану систему підстановок P. Виходячи з довільного слова α в алфавіті Σ розглядаються підстановки в тому порядку, в якому їх задано. Перша підстановка, ліва частина якої співпадає з деяким підсловом α, використовується для перетворення α, в яке замість першого входження лівої частини підстановки підставляється її права частина, внаслідок чого утворюється нове слово α_1. Далі, виходячи з цього нового слова α_1, процес повторюється, поки не закінчиться, ознаками чого є два випадки:

1) коли утворилося таке слово α_n, що жодна з лівих частин допустимих підстановок не є його підсловом;

2) коли при утворенні слова α_n використано останню підстановку.

*Означення II***.**
$$\begin{matrix}(q_0) & \Pi^*(S, S_1^0, S_2^0) & (l_0, r_0)\\ (q_1) & \Pi^*(S, S_1^1, S_2^1) & (l_1, r_1)\\ \dots & \dots & \dots\\ (q_n) & \Pi^*(S, S_1^n, S_2^n) & (l_n, r_n)\end{matrix}$$ – операторний алгоритм (1)

$П^*(S, S_1^1, S_2^1)$ – оператор (*S*-індивідна змінна; S_1, S_2 – деякі фіксовані слова).

Оператор $П^*(S, S_1^1, S_2^1)$ *реагує на слово S*, якщо S_1 входить в слово S. Оператор $П^*(S, S_1^1, S_2^1)$ не реагує на слово *S*, якщо S_1 не входить в *S*.

Нормальним алгоритмом Маркова називається алгоритм виду (1), в якому $r_i = q_{i+1}$, r_n – кінцева мітка ($r_n = Я$), $l_j \epsilon \{q_0, r_n\}$, $(i = \overline{0, n-1}; j = \overline{0, n})$.

Дані умови означають, що коли оператор $П^*(S, S_1^1, S_2^1)$ в команді (q_m) $П^*(S, S_1^m, S_2^m)$ (l_m, r_m) не реагує на слово *S*, то слово *S* не міняється і управління передається на наступну команду $(q_{m+!})$ $П^*(S, S_1^{m+1}, S_2^{m+1})$ (l_{m+1}, r_{m+1}). Якщо реагує, то над цим словом виконується операція $П^*(S, S_1^m, S_2^m)$ і управління передається або на початкову мітку q_0 або на кінцеву r_n – в залежності від того чи заключною є команда, оператор якої реагує на слово.

Команда (q_m) $П^*(S, S_1^m, S_2^m)$ (l_m, r_m) називається *заключною*, якщо $l_m = Я$. Скорочено позначається $S_1 \to \bullet S_2$.

Оператор називається *незаключним*, якщо $l_m \neq Я$ $(S_1 \to S_2)$.

Приклад нормального алгоритму

(q_0) $П^*(S, aba, b)$ (q_0, q_1)

(q_1) $П^*(S, aa, b)$ $(Я; q_2)$ – заключна команда

(q_2) $П^*(S, cbc, a)$ $(q_0, Я)$

Або можна записати в такому вигляді:

$$\Phi_0 : aba \to b$$
$$\Phi_1 : aa \to \bullet b \qquad (2)$$
$$\Phi_2 : cbc \to a$$

Робота нормального алгоритму**.**

Слово *S* послідовно зверху вниз проходить оператори нормального алгоритму. Оператори, які на *S* не зреагували, пропускають це слово без змін. Якщо слово *S* попадає на оператор. який на нього реагує, то даний оператор застосовується до цього слова і одержується нове слово S'. Якщо оператор, що зреагував, не є заключним, то слово S' іде на початок нормального алгоритму і тільки що вказана процедура застосовується до слова S'. Якщо оператор, що зреагував, є заключним, то алгоритм закінчує роботу і S' є його результатом. Якщо на *S* не зреагував жоден оператор нормального алгоритму, то алгоритм теж закінчує свою роботу і слово *S*, що не змінилося, є його результатом.

Приклад.

Проаналізуємо застосування алгоритму (2) до слова

$S = cbcab$

Такти	*i*	0	1	2	3	4	5	
Слова	S_i	*cbcab*	*cbcab*	*cbcab*	*aab*	*aab*	*bb*	кінець
Оператори	0_i	Φ_0	Φ_1	Φ_2	Φ_0	Φ_1		

Функція $f(x_1,\ldots,x_n)$ називається *нормально обчислювальною або М-функцією*, якщо існує нормальний алгоритм, що обчислює значення даної функції на кожному наборі аргументів з області визначення.

Принцип нормалізації Маркова.
Кожна обчислювальна функція є нормально обчислювальною.

Алгоритмічна система Тюрінга

В основі цієї системи лежить поняття машини Тюрінга.
Нехай маємо дві нескінчені системи символів $a_0, a_1,\ldots$ і $q_0, q_1,\ldots$Символи a_i називаються буквами, а q_i – мітками. Конкретний вигляд букв і міток не має значення. Єдине, що вимагається, – щоб сукупності букв і міток не перетиналися між собою.

Командою Тюрінга називається вираз вигляду (q, a) (q',a',r), де q,q' – мітки, a, a' – букви, r приймає одне з трьох чисел -1, 0, +1.

Пара (q, a) в команді (q, a) (q',a',r) називається *міткою Тюрінга* (або просто *Т-міткою*).

Програмою Тюрінга називається довільна скінчена сукупність $П_Т$ команд Тюрінга, в якій різні команди мають різні *Т*-мітки.

Тобто кожна команда в програмі $П_Т$ однозначно визначається за своєю міткою.

Машиною Тюрінга називається довільна пара $Т=(q_0, П_Т)$, де q_0 – мітка, $П_Т$ – програма Тюрінга.

Мітка q_0 називається *початковою міткою* машини $Т=(q_0, П_Т)$.

Приклад програми Тюрінга:

(q_0,a_0) $(q_0,a_2,+1)$
(q_0,a_1) $(q_0,a_1,-1)$
(q_0,a_2) $(q_1,a_1,+1)$ (1)
(q_1,a_1) $(q_2,a_0,0)$

Станом машини Тюрінга (або просто *Т-станом*) називається довільна нескінчена в обидві сторони послідовність вигляду:

$\ldots b_{-3}b_{-2}b_{-1}qb_0b_1b_2\ldots$ (2)

де b_i ($i=\ldots$, -2, -1, 0, 1, 2, …) – букви, q –мітка.
Пара (q, b_0) називається *Т-міткою* стану (2).

Нехай $Т=(q_0, П_Т)$ – довільна машина Тюрінга.

Стан (2) називається *початковим станом машини Тюрінга*, коли $q=q_0$.

Стан (2) називається *проміжковим станом машини Тюрінга*, коли (q,b_0) – проміжкова *Т*-мітка цієї машини.

Коли (q,b_0) – кінцева *Т*-мітка машини, то стан (2) називається *кінцевим станом даної машини.*

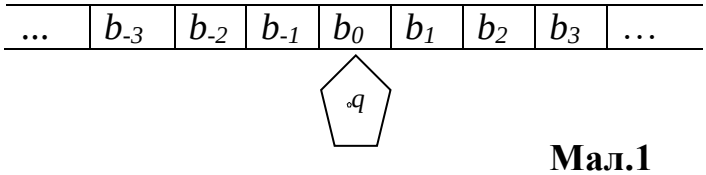
Визначимо функцію переходу $F_Т$ машини $Т=(q_0, П_Т)$.

Функція F_T кожен стан t вигляду (2) переводить в деякий стан $F_T(t)$.
Цей стан визначається так. Коли мітка (q,b_0) в стані (2) являється кінцевою міткою машини Тюрінга, то $F_T(t)=t$.

Нехай (q,b_0) – проміжкова Т-мітка машини $(q_0, П_T)$. Тоді в програмі $П_T$ машини $(q_0, П_T)$ знайдеться одна і тільки одна команда вигляду (q, a) (q',a',r). Під дією цієї команди t переходить в $F_T(t)$.
Залежно від r=+1, 0, -1 значення $F_T(t)$, відповідно, прийме такий вигляд:

$$\begin{aligned} &\ldots b_{-3}b_{-2}b_{-1}\,a'q'\,b_1b_2\ldots \\ &\ldots b_{-3}b_{-2}b_{-1}q'a'\,b_1b_2\ldots \qquad (3)\\ &\ldots b_{-3}b_{-2}q'b_{-1}a'\,b_1\,b_2\ldots \end{aligned}$$

Т-стани і функція F_T трактуються на мові “стрічки” і “голівки” машини Тюрінга.
Розглядається нескінчена в обидві сторони стрічка, яка розділена на комірки.
Стан (2) за допомогою “стрічки” і “голівки” представляється у вигляді:

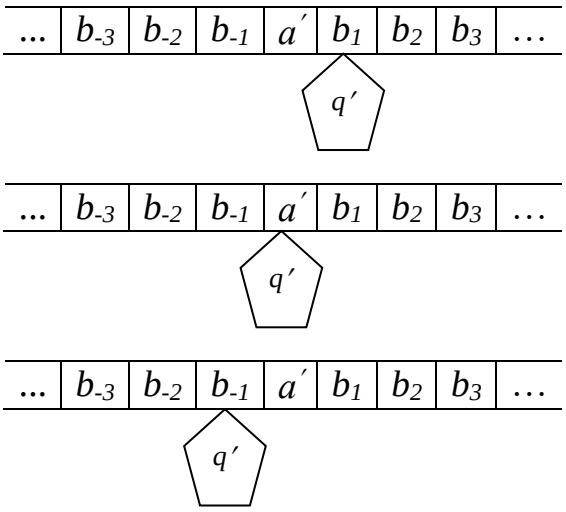


Мал.1

В кожній комірці стрічки в певний момент записана одна і тільки одна буква.
Голівка здійснює зчитування та запис букв на стрічці. В кожен момент голівка може сприймати одну і тільки одну комірку стрічки.
В процесі роботи машини Тюрінга голівка може пересуватись вліво і вправо по стрічці. Крім того, в голівці в кожен момент записана деяка мітка.

Послідовність … $b_{-2}\,b_{-1}\,b_0\,b_1\,b_2$ … називається *станом стрічки*, а мітка q – *станом голівки*. Стан голівки ще називається *внутрішнім станом*.

Стани (3), відповідно, представляються так:



Мал.2

Згідно мал.1 і мал.2 дію команди (q, a) (q', a', r) можна описати так.

В комірці, яку в стані мал.1 сприймає голівка, записується буква a', мітка q в голівці замінюється на мітку q' і залежно від r =+1, 0, -1 голівка пересувається на одну комірку вправо, залишається на місці і пересувається на одну комірку вліво.

Траєкторією машини Тюрінга називається послідовність станів t_0 , t_1 ,..., де t_0 – початковий стан даної машини і

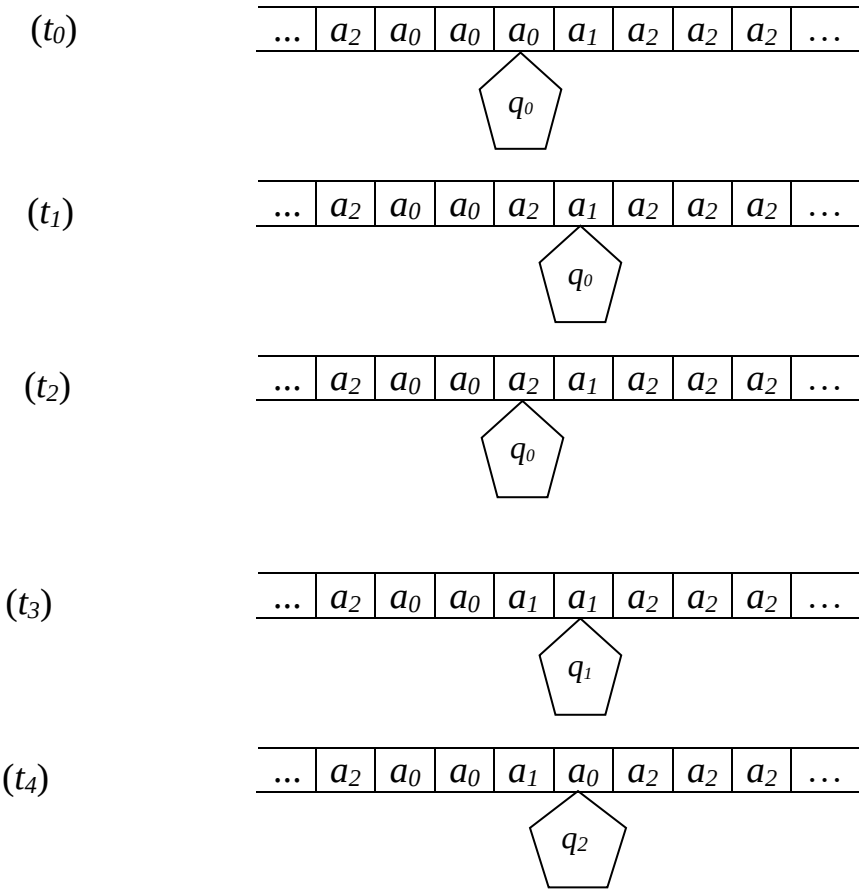
$$t_{i+1}=F_T(t_i) \quad (i=0, 1, 2, \ldots).$$

Як і для випадку операторних алгоритмів, індекси i при станах t_i будемо називати *тактами*.

Коли t_0 , t_1 ,..., t_{m-1} – проміжкові стани і t_m – кінцевий стан машини Тюрінга, то будемо говорити, що машина Тюрінга зупиняється на m-му такті.

Приклад.

Нехай T=(q_0, $П_T$), де $П_T$ має вигляд (1). Припустимо, що машина Тюрінга починає роботу зі стану вигляду ... a_2 a_0 a_0 q_0 a_0 a_1 a_2 a_2 .... Починаючи з цього стану машина Тюрінга опише таку траєкторію:



В даній траєкторії машина зупиняється на 4-му такті.

Побудуємо декілька машин Тюрінга, які назвемо *елементарними машинами Тюрінга*. Будемо позначати їх $T_i(q_0, ...)$, де q_0 – початкова мітка машини Тюрінга, а всі інші мітки, які входять у вираз $T_i(q_0, ...)$, вважаються кінцевими мітками даної машини.

Для всіх машин тимчасово зафіксуємо скінчений алфавіт $L_k=\{a_0, a_1, \dots a_{k-1}\}$ і будемо вважати, що на стрічці записуються тільки букви з даного алфавіту.

1. Машина $T_{-1}(q_0, q_1)$. Ця машина пересуває голівку на одну комірку вліво, не міняючи стану стрічки.

Програма для цієї машини має вигляд:

(q_0, a_i) $(q_1, a_i, -1)$ $(i=0, 1, ..., k-1)$.

2. Машина $T_{+1}(q_0, q_1)$. Ця машина пересуває голівку на одну комірку вправо, не міняючи стану стрічки.

Програма для цієї машини має вигляд:

(q_0, a_i) $(q_1, a_i, +1)$ $(i=0, 1, ..., k-1)$.

3. Машина $T_{-}(q_0, b, q_1)$, $b \in L_k$. Ця машина пересуває голівку вліво до тих пір, поки не зустріне букву b.

Програма для цієї машини має вигляд:

(q_0, a_i) (q_i', a_i, r_i) $(i=0, 1, ..., k-1)$, де $q_i' = q_0$, $r_i=-1$, коли $a_i \neq b$

і $q_i' = q_1$, $r_i=0$, коли $a_i=b$.

4. Машина $T_{+}(q_0, b, q_1)$, $b \in L_k$. Ця машина пересуває голівку вправо до тих пір, поки не зустріне букву b.

Програма для цієї машини має вигляд:

(q_0, a_i) (q_i', a_i, r_i) $(i=0, 1, ..., k-1)$, де $q_i' = q_0$, $r_i=+1$, коли $a_i \neq b$

і $q_i' = q_1$, $r_i=0$, коли $a_i=b$.

5. Машина $T_0(q_0, b, q_1)$, $b \in L_k$. Ця машина вписує в комірку, на яку вказує голівка, букву b.

Програма для цієї машини має вигляд:

(q_0, a_i) $(q_1, b, 0)$ $(i=0, 1, ..., k-1)$.

6. Машина $T_2(q_0, b, q_1, q_2)$, $b \in L_k$. Ця машина зчитує букву в комірці, яку сприймає голівка. Якщо ця буква являється буквою b, то управління передається на мітку q_1. Якщо буква, яка зчитується, відмінна від b,то управління передається на мітку q_2.

Програма для цієї машини має вигляд:

(q_0, a_i) $(q_i', a_i, 0)$ $(i=0, 1, ..., k-1)$, де $q_i' = q_1$, коли $a_i=b$

і $q_i' = q_2$, коли $a_i \neq b$.

7. Машина $T_{-}^{*}(q_0, a, b, q_5)$, $a, b \in L_k$. Нехай перед початком роботи цієї машини голівка в комірці зчитує букву a. Тоді ця буква a переноситься вліво до тих пір, поки перед нею (тобто лівіше неї) не з'явиться буква b.
8. Машина $T_{+}^{*}(q_0, a, b, q_5)$, $a, b \in L_k$. Ця машина виконує аналогічну роботу, що й машина $T_{-}^{*}(q_0, a, b, q_5)$, тільки переносить букву a вправо.

Функція f називається *T-функцією*, якщо існує машина Тюрінга, що обчислює дану функцію.

Гіпотеза Тюрінга. Класи обчислювальних і обчислювальних за Тюрінгом функцій збігаються.

Теорія рекурсивних функцій

В теорії рекурсивних функцій прийнято конструктивний підхід, основною рисою якого є те, що множина всіх об'єктів, які досліджуються (функцій), будується зі скінченого числа початкових об'єктів, тобто базису, за допомогою простих операцій, що називаються операторами.

Будемо будувати алгоритми для функції $f(x_1, x_2, \ldots, x_n)$, яка приймає цілочислові значення і залежить від цілочислових аргументів. Будемо розглядати множину натуральних чисел $0,1,\ldots$ і розглянемо тільки ті числові функції (функції n змінних), область визначення і область значень яких належать N.

Функція $f(x_1,...,x_n)$ називається *частково числовою n-місною функцією*, якщо $D_O(f) \subset N^n$ і $D_З(f) \subset N^n$, де

D_0 — область визначення

$D_з$ — область значень даної функції.

Якщо $D_O(f) = N^n$, то функція називається *всюди визначеною*.

Опишемо клас числових функцій, за допомогою яких будемо будувати обчислювальні функції.

Найпростіші функції:

1. $O(x) = 0$ (нуль-функція).
2. $S(x) = x+1$ (функція наступності, або функція додавання одиниці).
3. $I_m^n(x_1,...x_n) \equiv x_m$ (функція тотожності, або функція вибору аргументу, або функція введення фіктивних змінних).

Оператором суперпозиції S_m^n називається підстановка у функцію m змінних $f(x_1,...,x_m)$ m функцій $f_1(x_1,...,x_n),...,f_m(x_1,...,x_n)$ n одних і тих самих змінних.

В результаті одержуємо нову функцію n змінних

$\varphi(x_1,...,x_n) = F(f_1(x_1,...,x_n),...,f_m(x_1,...,x_n))$

Нехай V – деяка система функцій, визначених на N.

Кажуть, що функція f одержана із системи функцій V за схемою суперпозиції, якщо

$f(x_1,...,x_n) = \varphi(f_1(y_1,...,y_{l_1}),...,f_k(u_1,...,u_{l_k}))$

$y_i, u_j \in \{x_1,...x_n\}$

функції f_i або належать системі V, або є функціями вибору аргументу.

Наприклад, $f(x_1,x_2,x_3) = \varphi(x_1, f_1(x_1,x_2), f_2(x_1,x_2x_3))$.

Кажуть, що оператор *примітивної рекурсії* R^n визначає $(n+1)$-місну функцію $f(x_1,\ldots,x_n,y)$ через n–місну функцію $\varphi(x_1,\ldots,x_n)$ і $(n+2)$-місну функцію $\psi(x_1,\ldots,x_n,y,z)$ за схемою примітивної рекурсії, якщо

$$f(x_1,\ldots,x_n,0)=\varphi(x_1,\ldots,x_n)$$

$$f(x_1,\ldots,x_n,y+1)=\psi(x_1,\ldots,x_n,y,f(x_1,\ldots,x_n,y))$$

$f(x_1,\ldots,x_n,y)=R^n(\varphi,\psi)$ - позначення оператора примітивної рекурсії.

При $n=0$ функція f, що визначається, є одномісною і схема примітивної рекурсії має вигляд

$f(0)=a,\ f(y+1)=\psi(y,f(y))$.

З означення схеми примітивної рекурсії видно, що функція f визначається рекурсивно не тільки через інші функції φ і ψ, а й через значення функції f у попередніх точках (значення f у точці $y+1$ залежить від значення f у точці y).

Приклад.

Двомісна функція $f_+(x,y)=x+y$ задовольняє схему примітивної рекурсії:

$x+0=x=I_1^1(x)$;

$x+(y+1)=(x+y)+1=S(x+y)$

Функція f називається *примітивно-рекурсивною*, якщо вона може бути одержана з найпростіших за допомогою скінченого числа застосувань операторів суперпозиції та примітивної рекурсії.

Формальний вигляд цього означення:

1. $O(x), S(x), I_m^n(x_1,\ldots,x_n)\ \ (\forall n, m\in N, m=1,\ldots,n)$ є примітивно-рекурсивними.
2. Якщо функції $f(x_1,x_2,\ldots,x_m), f_1(x_1,x_2,\ldots,x_n),\ldots,f_m(x_1,x_2,\ldots,x_n)$ примітивно-рекурсивні, то функції $S_m^n(f,f_1,\ldots,f_m)$ теж примітивно-рекурсивні.
3. Якщо функції $\varphi(x_1,\ldots,x_n)$ і $\psi(x_1,\ldots,y,z)$ примітивно-рекурсивні, то функція $R^n(\varphi,\psi)$ – примітивно-рекурсивна.

Жодних інших примітивно-рекурсивних функцій немає.

Схема мінімізації.

Нехай задано $\varphi(x_1,\ldots,x_n,u)$ на множині N.

Зафіксуємо довільний набір $x_1,\ldots,x_n$. Шукаємо таке z, що

$\varphi(x_1,\ldots,x_n,z)=0$

і $\forall y\ (y<z)\ \varphi(x_1,\ldots,x_n,y)\neq 0,\ \varphi(x_1,\ldots,x_n,y)\neq\pi$

Дані умови через функцію $\varphi(x_1,\ldots,x_n,u)$ однозначно визначають нову функцію $z=f(x_1,\ldots,x_n)$ і кажуть, що функція $f(x_1,\ldots,x_n)$ одержується з функції $\varphi(x_1,\ldots,x_n,u)$ за схемою мінімізації і позначається $f(x_1,\ldots,x_n)=\mu_u(\varphi(x_1,\ldots,x_n,u)=0)$.

Якщо такого z, що задовольняє вищезгаданим умовам не існує, то покладається $f(x_1,\ldots,x_n)=\pi$.

Якщо $\varphi(x_1,\dots,x_n,u)$ є всюди визначеною на множині N, то необхідне z представляє собою найменший розв'язок рівняння

$\varphi(x_1,\dots,x_n,u)=0$ при фіксованому наборі $x_1,\dots,x_n$.

В даному випадку $f(x_1,\dots,x_n,u)\neq\pi$ тоді і тільки тоді, коли рівняння $\varphi(x_1,\dots,x_n,u)=0$ має хоча б один розв'язок відносно змінної u.

Коли ж функція $\varphi(x_1,\dots,x_n,u)$ не є всюди визначеною на N, то може виникнути ситуація, при якій рівняння $\varphi(x_1,\dots,x_n,u)=0$ має розв'язок, але все-таки $f(x_1,\dots,x_n)=\pi$. Це має місце, коли

$\varphi(x_1,\dots,x_n,z)=0$

$\forall y\ (y<z)\ \varphi(x_1,\dots,x_n,y)\neq 0$

і $\exists y\ (y<z)\ \varphi(x_1,\dots,x_n,y)=\pi$.

Будемо говорити, що функція $f(x_1,\dots,x_n)$ одержується із системи V за схемою мінімізації, якщо у V існує така функція $\varphi(x_1,\dots,x_n,u)$, що

$$f(x_1,\dots,x_n)=\mu_u(\varphi(x_1,\dots,x_n,u)=0).$$

Функція називається *частково-рекурсивною*, якщо вона може бути одержана з найпростіших функцій за допомогою скінченого числа застосувань операторів суперпозиції, примітивної рекурсії та мінімізації.

Загальнорекурсивними називаються частково-рекурсивні функції, які є всюди визначені на множині N.

Теза Черча.

Клас алгоритмічно обчислювальних частково числових функцій збігається з класом усіх частково-рекурсивних функцій.

Дана гіпотеза доводиться багаторічною математичною практикою, але математично її обґрунтувати неможливо, оскільки в неї входить інтуїтивне поняття обчислювальності.

Твердження. Система функцій $\{O(x), S(x), I_1^1(x)\}$ є повною системою відносно операцій суперпозиції, примітивної рекурсії та мінімізації у множині алгоритмічно обчислювальних частково числових функцій.

Теорема. Система функцій $\{O(x), S(x)\}$ є повною системою відносно операцій суперпозиції, примітивної рекурсії та мінімізації у множині алгоритмічно обчислювальних частково числових функцій.

Доведення.

Розглянемо схему, яка визначає $I_1^1(x)$ через $O(x)$ і $S(x)$:

$$I_1^1(0)=O(x)=0$$

$$I_1^1(x+1)=S(I_1^1(x))$$

це є схема примітивної рекурсії. Отже, система функцій $\{O(x), S(x)\}$ є повною.

Алгоритмічна система Кліні

Нехай $K_0=\{x+1, 0\}$ – система функцій.

Функція $f(x_1,\ldots,x_n)$ називається *функцією Кліні, або К-функцією*, якщо вона може бути одержана скінченим числом застосувань схем суперпозиції, примітивної рекурсії та мінімізації до функцій системи K_0.

Скінчена послідовністю $f_1,\ldots,f_n$ називається *К-послідовністю*, якщо для кожного $i(0<i<n)$ виконується хоча б одна з таких умов :

1) $f_i \in K_0$;
2) f_i одержується з функцій системи $\{f_0,f_1,\ldots f_{i-1}\}$ або за схемою суперпозиції, або за схемою примітивної рекурсії, або за схемою мінімізації.

Функція f_n називається *кінцем К-послідовності*.

Уточнимо поняття *К*-функції.

Функція f називається *К-функцією*, якщо вона є кінцем хоча б однієї *К*-послідовності.

З означення *К*-послідовності випливають такі леми:

Лема1. Кожна послідовність $f_0,\ldots,f_i$ $(0<i\le n)$ *К*-послідовності $f_1,\ldots,f_n$ теж являється *К-послідовністю*.

Наслідок. Функція $f(x_1,\ldots,x_n)$ є *К*-функцією тоді і тільки тоді коли вона є членом хоча б однієї *К*-послідовності.

Лема2. Якщо послідовності

$f_{00},\ldots, f_{0n_0}$,

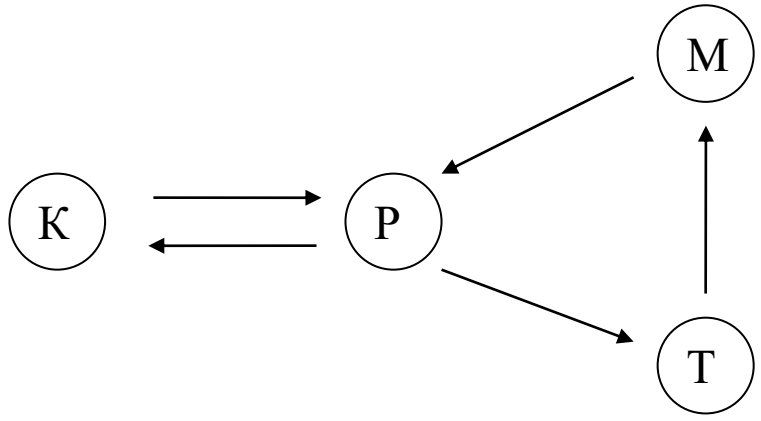
$f_{10},\ldots, f_{1n_1}$,

$f_{0m},\ldots, f_{mn_m}$ є *К*-послідовностями, то й послідовність

$f_{00},\ldots, f_{0n_0}, f_{10},\ldots, f_{1n_1}, f_{0m},\ldots, f_{mn_m}$ теж є *К*-послідовністю.

Лема3. Нехай U – система *К*-функцій. Нехай функція f одержується з системи U або за схемою суперпозиції, або за схемою мінімізації, або за схемою примітивної рекурсії. Тоді f являється *К*-функцією.

Узагальнена схема еквівалентності алгоритмічних систем



К – алгоритмічна система Кліні,
Р – рекурсивні функції,
М – алгоритмічна система Маркова,
Т – алгоритмічна система Тюрінга.

Завдання для самостійної роботи

1. Інтерпретувати дану програму Тюрінга на мові „стрічки”, якщо початковим є стан ... $a_0\ a_0\ a_2\ q_0\ a_0\ a_1\ a_1\ a_2$ ...
   $(q_0, a_0)\ (q_1, a_0, +1)$
   $(q_1, a_1)\ (q_1, a_1, -1)$
   $(q_1, a_0)\ (q_2, a_1, -1)$
   $(q_2, a_2)\ (q_3, a_1, +1)$.

2. В який стан переведе дана послідовність елементарних машин Тюрінга такий стан ... $q_0\ a_0\ a_1\ a_1\ b\ a_0\ c$ ...
   $T_{+1}\ (q_0, q_1)$
   $T_0\ (q_1, a_1, q_2)$
   $T_{-1}\ (q_2, q_2)$
   $T_+\ (q_2, b, q_3)$.

3. Записати дану програму у вигляді послідовності елементарних машин Тюрінга, якщо початковим є стан ... $a_0\ q_0\ a_2\ a_1$ ...
   $(q_0, a_2)\ (q_1, a_1, -1)$
   $(q_1, a_0)\ (q_2, a_2, +1)$
   $(q_2, a_1)\ (q_3, a_1, 0)$.

4. Дано програму
   $(q_0, a_1)\ (q_0, a_2, +1)$
   $(q_0, a_0)\ (q_1, a_2, +1)$

(q_1, a_0) $(q_2, a_0, 0)$,

S_2 – множина слів над алфавітом $L_2=\{a_0, a_1\}$, $s=a_1\ a_0\ a_0$. Знайти функцію переходу цього слова.

5. Записати дану послідовність елементарних машин Тюрінга у вигляді програми Тюрінга
   $T_0\ (q_0, b, q_1)$
   $T_{+1}\ (q_1, q_1)$
   $T_{+1}\ (q_1, q_2)$
   $T_0\ (q_2, c, q_3)$
   $T_+\ (q_3, a_2, q_4)$.

6. Записати траєкторію у вигляді „стрічки”, яку одержимо, застосувавши дану послідовність елементарних машин Тюрінга до стану
   $\ldots a_0\ a_0\ b a_1\ q_0\ a_2\ a_1\ d\ a_2 \ldots$
   $T_0\ (q_0, a_0, q_1)$
   $T_+^*\ (q_1, a_0, d, q_2)$
   $T_+\ (q_2, a_2, q_3)$
   $T_{-1}\ (q_4, q_4)$
   $T_2\ (q_4, b, q_5, q_6)$.

7. В слові $s=\ a_0\ a_1\ a_2\ a_1\ a_1\ a_1\ a_0\ a_1$ замінити всі букви a_1 на a_3. Написати програму Тюрінга. Початкова мітка знаходиться на букві a_2.
8. В слові $s=\ a_2\ a_1\ a_0\ a_2\ a_2\ a_1\ a_2\ a_1$ замінити всі букви a_2 на a_3. Написати послідовність елементарних машин Тюрінга. Початкова мітка знаходиться на початку слова.
9. В слові $s=$*гідроелектростанція* замінити всі глухі приголосні на букву „о”, а всі дзвінкі приголосні – на „а”. Написати послідовність елементарних машин Тюрінга. Початкова мітка знаходиться на початку слова.
10. Зі слова *геометрія* утворити слово *межа*. Написати послідовність елементарних машин Тюрінга та у вигляді програми Тюрінга. Початкова мітка знаходиться на букві *m*.
11. Записати траєкторію і визначити результат застосування даного нормального алгоритму до слова $s=$ *cbacab*
    $(q_0)\ \Pi^*\ (s, aac, a)\ (q_0, q_1)$
    $(q_1)\ \Pi^*\ (s, cba, b)\ (q_1, q_2)$
    $(q_2)\ \Pi^*\ (s, ab, c)\ (Я, q_3)$

12. Дано нормальний алгоритм. Знайти слово, яке одержиться зі слова
    $s=$ *kbladmab*
    $(q_0)\ \Pi^*\ (s, kl, b)\ (q_0, q_1)$
    $(q_1)\ \Pi^*\ (s, lm, ml)\ (q_1, q_2)$
    $(q_2)\ \Pi^*\ (s, mmb, a)\ (Я, q_3)$

(q_3) П* (s ,db ,mb) (q_3 , q_4)
(q_4) П* (s ,mab ,lla) (q_4 , $Я$).

13. Написати нормальний алгоритм Маркова, який в будь-якому слові замінює всі букви a_2 на a_0 .
14. Написати нормальний алгоритм Маркова, який впорядковує слово з чотирьох елементів по зростанню.
15. Перевірити, чи задовольняє двомісна функція додавання $f_+(x,y)=x+y$ схему примітивної рекурсії.
16. Перевірити, чи задовольняє двомісна функція множення $f_\times(x,y)=x\times y$ схему примітивної рекурсії.
17. Перевірити, чи задовольняє двомісна функція піднесення до степеня $f_s(x,y)=x^y$ схему примітивної рекурсії.

Завдання для індивідуальної роботи

1. Типи логік.
2. Приклади алгоритмічно нерозв′язних проблем.
3. Принцип резолюції.
4. Основні методи математичних доведень.
5. Машина Поста.
6. Метод дедукції.
7. Методи автоматичного доведення теорем.
8. Гірлянди, дерева, піддерева.
9. Моделі і мови представлення знань.
10. Кляузи Хорна.
11. Підхід Ербрана до автоматичного доведення теорем.
12. Клаузальні форми логіки.
13. Формальні граматики.
14. Загальна характеристика методів представлення знань.
15. Проблема синтезу довільної програми.
16. Логічна модель представлення знань (логічне програмування).

Алгебраїчні структури

Алгеброю A називається сукупність $\langle\ \rangle$ множини M з заданими в ній операціями $S = \{f_{11}, f_{12},..., f_{1n_1}, f_{21}, f_{22},..., f_{2n_2},..., f_{m1}, f_{m2},...,..., f_{mn_m}\}$, $A=\langle M, S\rangle$, де множина M – *носій, S – сигнатура* алгебри. Перший нижній індекс у ідентифікатора операції вказує її *арність.*

Зауваження**.** Для ідентифікації єдиного цілого, що містить об'єкти, які мають різну математичну будову, наприклад, множини і операції в ній, вживається термін *сукупність* і позначають його кутовими дужками.

Розглянемо *фундаментальні алгебри*.

Алгебра виду $\langle M, f_2\rangle$ називається *групоїдом*.

Якщо f_2 – операція типу множення ($\times$), то групоїд називається *мультиплікативним;* якщо f_2 – операція типу додавання (+), то *адитивним.*

Нехай $A=\langle M, f_2\rangle$ – групоїд; позначимо операцію f_2 як $\circ$.

Тоді елемент $e \in M$ називається *правим нейтральним елементом* групоїда A, якщо для довільного $m \in M$ виконується рівність $m \circ e = m$; *елемент* $e \in M$ групоїда $A=\langle M,\circ\rangle$ називається *лівим нейтральним елементом*, якщо для всіх $m \in M$ виконується рівність $e \circ m = m$.

В цих визначеннях використовувались вирази "всі елементи", "всякий елемент".

Замість слів "все" і "всякий" ("довільний", "будь-який") можна використовувати символ $\forall$ (перевернута буква А – перша буква англійського слова All – все).

Якщо елемент e, $e \in M$, групоїда $A=\langle M,\circ\rangle$ являється одночасно лівим і правим нейтральним елементом, то його називають *двостороннім нейтральним елементом* або просто *нейтральним елементом*.

Жоден групоїд не може мати більше одного нейтрального елемента. Дійсно, якщо

$$m \circ e = e \circ m = m \quad \text{і} \quad m \circ e' = e' \circ m = m$$

справедливо для всіх $m \in M$, то

$$e' = e' \circ e = e.$$

Якщо групоїд $\langle M,\circ\rangle$ мультиплікативний, то нейтральний елемент називається *одиницею* і позначається 1; якщо адитивний, то нейтральний елемент називається *нулем* і позначається 0.

Групоїд $A=\langle M,\circ\rangle$ називається *ідемпотентним*, якщо його сигнатура задовольняє закону ідемпотентності

$$(\forall m \in M)(m \circ m = m).$$

Групоїд $\langle M,\circ\rangle$, сигнатура якого задовольняє закону комутативності

$$(\forall x, y \in M)(x \circ y = y \circ x),$$

називається *комутативним* або *абелевим*.

Групоїд $\langle M,\circ\rangle$, в якому виконується закон асоціативності

$(\forall x,y,z\in M)(x\circ(y\circ z)=(x\circ y)\circ z)$,

називається *асоціативним* або *напівгрупою*.

Означення 1. Напівгрупа $\langle M,\circ\rangle$, в якій виконуються обернені операції: для довільних $a,\ b\in M$ кожне з рівнянь $a\circ x=b,\ y\circ a=b$ має єдиний розв'язок, називається *групою.*

Означення 2. *Групою* називається непорожня множина M об'єктів довільної природи, якщо на ній задана бінарна алгебраїчна операція $(M\times M\to M)$, яка задовольняє наступним умовам:

1) $\forall a,b,c\quad (a\circ b)\circ c=a\circ(b\circ c)$ (асоціативність);
2) $\exists e\in M\quad \forall a\in M\quad e\circ a=a\circ e=a$ (існування одиниці);
3) $\forall a\in M\quad \exists a^{-1}\in M\quad aa^{-1}=a^{-1}a=e$ (існування оберненого елемента).

Підмножина H групи M називається *підгрупою*, якщо H сама є групою відносно операції, заданої на M.

Приклади груп.

1. Відмінні від нуля раціональні числа з операцією множення.
2. Додатні раціональні числа з операцією множення.
3. Числа +1 і –1 по множенню.
4. Число 0 утворює групу по множенню.
5. Цілі числа (операція додавання).
6. Раціональні числа (операція додавання).
7. Число 0 утворює групу по додаванню.

Проілюструємо поняття групи на прикладі *групи підстановок.*

Перестановки деяких елементів – це всі можливі способи, якими ці елементи (частіше – числа) можна вистроїти в ряд.

Число перестановок з n елементів дорівнює $n!$.

Підстановка – це операція, яка міняє порядок елементів в перестановці або

підстановка n-ї степені – це взаємно-однозначне відображення множини із n елементів на себе.

Для задання підстановки необхідно:

1) задати область визначення, тобто всю сукупність елементів, над якими проводиться підстановка;
2) задати алгоритм підстановки, тобто для кожного елемента з області визначення вказати елемент, в який він перейде під дією підстановки, причому різні елементи мають переходити в різні.

Введемо операцію множення $\times$ над підстановками.

Добутком підстановок називається підстановка, яка одержується в результаті послідовного виконання спочатку першої, а потім другої із перемножуваних підстановок.

Наприклад, якщо $a=\begin{pmatrix} x_1 & x_2 & x_3 \\ x_2 & x_1 & x_3 \end{pmatrix}$, $b=\begin{pmatrix} x_1 & x_2 & x_3 \\ x_1 & x_3 & x_2 \end{pmatrix}$, то

$$a\times b=\begin{pmatrix} x_1 & x_2 & x_3 \\ x_3 & x_1 & x_2 \end{pmatrix}.$$

Множення підстановок не є комутативним, але є асоціативним.

Одиницею групи підстановок є тотожня підстановка *I,* тобто якщо кожен елемент переходить сам в себе

$$I=\begin{pmatrix} x_1 & x_2 & x_3 & x_4 \\ x_1 & x_2 & x_3 & x_4 \end{pmatrix}.$$

Для кожної підстановки P існує обернена P^{-1}, яка знаходиться заміною верхнього рядка на нижній. Справедливо співвідношення:

$$P\times P^{-1}=P^{-1}\times P=I.$$

Отже, множина підстановок з введеною операцією множення є групою (неабелевою).

Циклічні підгрупи.

Нехай G – довільна група, g – довільний елемент із G. Утворимо множину $\langle g\rangle=\{\, g^n / n\in Z\,\}$. Легко перевірити, що це група. Вона має назву *циклічна підгрупа групи групи G, утворена елементом g.*

Якщо група G нескінченого порядку, то й елемент g має нескінчений порядок і ця множина має вигляд:

$$\langle g\rangle=\{\ldots g^{-2}, g^{-1}, 1, g, g^2, \ldots\}.$$

Якщо $\langle g\rangle$ складається з n елементів, тоді вона має вигляд:

$$\langle g\rangle=\{1, g, g^2, \ldots, g^{n-1} / g^n=1\}.$$

Тоді елемент g називається *елементом n-го порядку.*

Порядок елемента g – це найменше натуральне число n, для якого $g^n=1$.

Якщо ж такого натурального n не існує, то g – елемент нескінченого порядку.

Приклади.

1. Множина комплексних чисел по множенню.
2. Число –1 має порядок 2, тобто $\langle -1\rangle=\{1,-1\}$.

3. Число i має порядок 4, тобто $\langle i \rangle = \{1, i, -i, -1\}$.
4. Число 2 є елементом нескінченого порядку, тобто $\langle 2 \rangle = \left\{ ..., \frac{1}{4}, \frac{1}{2}, 1, 2, 4, ... \right\}$
5. $\varepsilon = \cos\frac{2\pi}{3} + i\sin\frac{2\pi}{3}$ – елемент 3-го порядку, тобто $\langle \varepsilon \rangle = \{1, \varepsilon, \varepsilon^2\}$.

Група G називається *циклічною,* якщо вона співпадає з однією із своїх циклічних підгруп.

Теорема. Кожна підгрупа циклічної групи циклічна; породжується тим елементом, який має найменшу степінь.

Суміжні класи. Групи за підгрупою.

Нехай H – підгрупа групи G, тоді для довільного елемента $g \in G$ можна утворити

$$gH = \{gh / h \in H\}.$$

Ця множина називається *лівим суміжним класом G/H* , аналогічно множина Hg називається *правим суміжним класом*.

Суміжні класи мають такі властивості:
1. $gH = H \Leftrightarrow g \in H.$
2. $aH = bH \Leftrightarrow a^{-1}b \in H.$
3. Кожен елемент суміжного класу можна вибрати представником суміжного класу:
$$\forall b \in aH \quad aH = bH.$$
4. Якщо група G скінчена, то кожен суміжний клас містить стільки елементів, який порядок підгрупи H.

5.Теорема Лагранжа. В скінченій групі G порядок підгрупи ділить порядок групи.

Число різних суміжних класів групи G по підгрупі H називається *індексом підгрупи H.*

Якщо група G має порядок n, то підгрупи порядку 1 і n називаються *тривіальними підгрупами.*

Нормальні підгрупи, фактор-групи, гомоморфізми.

Підгрупа H групи називається *нормальною*, якщо $\forall g \in G$
$gH = Hg$ (1).

Позначають: $H \lhd G$.

Умова (1) еквівалентна такій умові:

$\forall g \in G \quad g^{-1} Hg = Hg$ (2).

Якщо $H \lhd G$, то $aHbH=abH$ (3).

Звідси випливає, що добуток двох суміжних класів також є суміжним класом.

Множина всіх суміжних класів групи G по H відносно операції (3) утворює групу. Ця група називається фактор-групою групи G по H . Позначають G/H.

$$G/H = \{1 \cdot H, g_2 H, ..., g_s H\}, \; s = |G : H|.$$

У G /H одиничний елемент 1 H , обернений $(gH)^{-1} = g^{-1}H$.

Приклади.

1. Група трикутника $T= \{1, a, a^2, b, ab, a^2b / a^3=1, b^2=1, ba=a^2b, ba^2=ab.\}$
   $H= \{1, a, a^2\}$ – нормальна підгрупа групи T,
   $T=H \cup bH=H \cup Hb$
   $T/H= \{1\,H, bH / (bH)^2=1\,H\}$ – циклічна група 2-го порядку.
2. $G=Z, H=5Z.$
   $G= 0+H \cup 1+H \cup 2+H \cup 3+H \cup 4+H,$
   $G/H = \{\bar{0}, \bar{1}, \bar{2}, \bar{3}, \bar{4}\} \cong Z_5,$
   $\bar{3} + \bar{3} = (3+H)+(3+H)=6+H=1+H=1.$

Завдання для самостійної роботи.

1. Довести, що множина всіх дійсних чисел утворює групу по додаванню і не утворює групу по множенню.
2. Довести, що множина всіх дійсних чисел, які не рівні нулю, утворює групу по множенню і не утворює групу по додаванню.
3. Довести, що додатні дійсні числа утворюють групу по множенню і не утворюють групу по додаванню.
4. Довести, що невід'ємні дійсні числа не утворюють групу ні по додаванню, ні по множенню.
5. Довести, що комплексні числа відмінні від нуля утворюють групу по множенню і не утворюють групу по додаванню.
6. Довести, що комплексні числа з модулем рівним одиниці утворюють групу по множенню і не утворюють по додаванню.
7. Довести, що комплексні числа з модулем більшим від одиниці не утворюють групу ні по множенню, ні по додаванню.
8. Довести, що цілі додатні степені двійки (числа 2, 4, 8, ...) не утворюють групу ні по множенню, ні по додаванню.

9. Довести, що всі степені двійки (з цілими додатними, цілими від'ємними і нульовим показником) утворюють групу по множенню і не утворюють групу по додаванню.
10. Довести, що числа виду $a+bi$, де a,b – цілі, $i=\sqrt{-1}$ утворюють групу по додаванню і не утворюють групу по множенню.
11. Перевірити, чи утворює групу множина всіх цілих чисел з операцією піднесення до степеня $h(a,b)=a^b$.

Означення. Відображення f: $G \xrightarrow{\text{"в"}} G'$ називається *гомоморфізмом*, якщо

$$\forall x, y \in G \quad f(xy) = f(x)f(y).$$

Ядром гомоморфізму називається множина всіх елементів G, що відображаються в одиничний елемент групи G'. Позначається $Ker\ f = \{y \in G \,/\, f(y) = 1'\}$.

Ядро гомоморфізму завжди є нормальним дільником групи.

Основна теорема про гомоморфізм**.** Якщо f: $G \xrightarrow{\text{на}} G'$ і $H=Ker\ f$, то $G/H \cong G'$.

Теорема про відповідність**.** Якщо відображення f: $G \xrightarrow{\text{на}} G'$ гомоморфне, то існує взаємно-однозначна відповідність між всіма підгрупами групи G' і тими підгрупами групи G, що містять ядро гомоморфізму. При цьому якщо групі $A' \subset G'$ відповідає $A \subset G$, то $G/A \cong G'/A'$.

Теорема про природній гомоморфізм**.** Кожна нормальна підгрупа H групи G є ядром природнього гомоморфізму τ: $G \to G/H$, що задається формулою $\tau(g) = gH \quad \forall g \in G$.

Лема**.** Якщо $A \lhd G$, B – будь-яка підгрупа G, то AB буде підгрупою групи G.

Теорема про ізоморфізм**.** Якщо $A \lhd G$, $B \subseteq G$, то $A \cap B \lhd B$ і $AB/A \cong B/A \cap B$.

Приклади.

1. $f: C^* \xrightarrow{\text{на}} R_+^*$, $f(z) = |z|$, $z \in C^*$; $f(xy) = |xy| = |x||y| = f(x)f(y)$. Це гомоморфізм. $Ker\ f = W$; $C^*/W \cong R_+^*$
2. $f: GL(n,C) \xrightarrow{\text{на}} C^* \qquad \forall A \in GL(n,C) \qquad f(A) = \det A$;
   $f(AB) = \det(AB) = \det A \det B = f(A)f(B)$
   Це гомоморфізм; $Ker\ f = SL(n,C)$; $GL(n,C)/SL(n,C) \cong C^*$.

Класи спряжених елементів.

Означення. Нехай $M \subseteq G$. *Централізатором множини* M в групі G називається множина

$$C_G(M) = \{g \in G / gm = mg \ \forall m \in M\}.$$

Централізатор завжди утворює підгрупу групи G.

Означення. *Нормалізатором підмножини* M групи G називається множина

$$N_G(M) = \{g \in G / gM = Mg\}.$$

Нормалізатор також становить підгрупу групи G.

Легко бачити, що $C_G(M) \subseteq N_G(M)$. Якщо M – підгрупа групи G, то $M \subseteq N_G(M)$ і навіть $M \lhd N_G(M)$. Крім того, $N_G(M)$ є найбільшою підгрупою групи G, в якій підгрупа M є нормальним дільником.

Твердження. $C_G(M) \lhd N_G(M)$.

Приклади.

1. Нехай $G = GL(2,R)$, $a = a = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$. Знайти $C_G(a)$. Для цього потрібно знайти такі матриці, що $\begin{pmatrix} x & y \\ z & t \end{pmatrix}\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} x & y \\ z & t \end{pmatrix}$. Звідси випливає система рівнянь:

$$\begin{cases} x = x + z \\ x + y = y + t \\ z = z \\ z + t = t \end{cases} \Rightarrow \begin{cases} z = 0 \\ x = t \end{cases}.$$

Отже, $C_G(a) = \left\{ \begin{pmatrix} x & y \\ 0 & x \end{pmatrix} \middle| x, y \in R, x \neq 0 \right\}$.

3. Знайти нормалізатор підгрупи $H = \left\{ \begin{pmatrix} 1 & w \\ 0 & 1 \end{pmatrix} \middle| w \in R \right\}$. Використовуємо співвідношення

$\begin{pmatrix} x & y \\ z & t \end{pmatrix}\begin{pmatrix} 1 & w \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & w \\ 0 & 1 \end{pmatrix}\begin{pmatrix} x & y \\ z & t \end{pmatrix}$ $\forall w \in R$. Одержуємо систему рівнянь:

$$\begin{cases} x = x + wt \\ z = z \\ xw + y = y + wt \\ zw + t = t \end{cases} \Rightarrow \begin{cases} wz = 0 \\ xw = wt \Rightarrow x = t \\ zw = 0 \end{cases}.$$

Отже, $N_G(H) = \left\{ \begin{pmatrix} x & y \\ 0 & x \end{pmatrix} \middle| x, y \in R, x \neq 0 \right\}$.

Означення. Елементи a і b групи G називаються *спряженими*, якщо існує такий елемент g із G, що: $g^{-1}ag = b$. Позначають: $a \sim b$.

Властивості.

1. $a \sim a$ (рефлексивність).
2. $a \sim b = a \sim b$ (симетричність).
3. $a \sim b,\ b \sim c \Rightarrow a \sim c$ (транзитивність).

Звідси випливає, що множина всіх елементів групи G розбивається на класи спряжених між собою елементів, причому різні класи не перетинаються.

Позначається: K_a – клас елементів групи G, спряжених з a.

Приклади.

1. G – абелева група. Тоді для довільного елемента a із G $K_a = G$.
2. $G = T = \{1, a, a^2, b, ab, a^2b \mid a^3 = 1, b^2 = 1, ba = a^2b, ba^2 = ab\}$. Знайти K_a.

$K_1 = \{g^{-1}1g \mid g \in G\} = \{1\}$.

$1^{-1}a1 = a$, $a^{-1}aa = a$, $a^{-2}aa^2 = a$; $b^{-1}ab = bab = a^2bb = a^2$;

$(ab)^{-1}a(ab) = b^{-1}a^{-1}ab = a^2 \Rightarrow K_a = \{a, a^2\}$; $K_b = \{b, ab, a^2b\}$

$a^{-1}ba = a^2ba = a^2a^2b = ab$

$a^{-2}ba^2 = aab = a^2b$.

Теорема про порядок класу K_a . Порядок класу K_a дорівнює індексу нормалізатора в групі G:

$|K_a| = |G : N_G(a)|$.

Наслідок. Порядок класу ділить порядок групи.

Означення. Підгрупи A і B групи G називаються *спряженими,* якщо $\exists g \in G \quad g^{-1}Ag = B$.

Множина всіх підгруп розпадається на класи спряжених підгруп.

Теорема про порядок класу K_A. $|K_A| = |G : N_G(A)|$.

Теорема. Підгрупа A групи G буде нормальною тоді і тільки тоді, коли вона спряжена лише з собою, тобто $|K_A = 1|$.

Теорема. Підгрупа A групи G буде нормальною в групі G тоді і тільки тоді, коли з кожним елементом a із A містяться в підгрупі A всі елементи, спряжені з a.

Центр і комутант

Означення. *Центром групи G* називається множина тих елементів групи G , які перестановні з кожним елементом групи G. Позначається $\zeta(G)$. А це фактично:

$\zeta(G)= C_G(G)$.

Ясно, що $\zeta(G)$ підгрупа і $\zeta(G) \lhd G$, бо $\forall g \in G$ $g\zeta(G)=\zeta(G)g$. Якщо G – абелева, то $\zeta(G)=G$. Для неабелевих груп $\zeta(G) \neq G$

Приклади.

1. $G= GL(n,K)$, K – поле. Тоді
$$\zeta(G)=\left\{ \begin{pmatrix} \lambda & & 0 \\ & \dots & \\ 0 & & \lambda \end{pmatrix} \middle| \lambda \in K^* \right\}.$$
2. $\zeta(T)=\{1\}$.
3. $\zeta(S_n)=1$, $n>2$.

Означення. *Комутатором* двох елементів a і $b \in G$ називається добуток $a^{-1}b^{-1}ab = [a,b]$.

Звідси $ab=ba[a,b]$.

Означення. Підгрупа групи G, породжена всіма комутаторами, називається *комутантом* групи G. Позначається G'.

$G' = < [a,b] \ | a,b \in G >$.

Кільце. Тіло. Поле.

Означення1. Алгебра $\langle M,\times,+\rangle$, яка по множенню являється мультиплікативним групоїдом, по додаванню – абелевою групою, причому множення зв'язано законами дистрибутивності

$a\times(b+c) = a\times b+a\times c$,

$(b+c)\times a = b\times a+c\times a$,

називається *кільцем.*

Означення 2. Непуста множина K елементів довільної природи, на якій введено дві бінарні алгебраїчні операції додавання і множення, називається *кільцем,* якщо виконуються такі аксіоми кільця:

1. $a+(b+c)=(a+b)+c$;
2. $a+b=b+a$;
3. $\exists 0 \in K \ \forall a \in K : a+0=a$ (0 називається *нулем кільця К*);
4. $\forall a \in K \ \exists(-a) \in K : a+(-a)=0$ (*-а* називається *протилежним* до *а*),

тобто по відношенню до додавання K має бути комутативною групою.

5. $a(b+c)=ab+ac$;
6. $(a+b)c=ac+bc$.

Якщо виконується

7. $a(bc)=(ab)c$, то кільце називається *асоціативним*.

Якщо ще виконується

8. $ab=ba$, то кільце називається *комутативним*.

Якщо ще виконується

9. $\exists e \in K \ \forall a \in K \ ae=ea=a$, то e називається *одиницею К,* а кільце K називається *кільцем з одиницею*.

Означення. Кільце, в якому всі відмінні від нуля елементи складають групу по множенню, називається *тілом.*

Означення 1. Тіло, в якого мультиплікативна група абелева, називається *полем.*

Означення 2. *Поле* – це множина М з двома бінарними операціями + і $*$, такими що:

1. $(a+b)+c=a+(b+c)$; асоціативність додавання;
2. $\exists 0 \in M \quad a+0=0+a=a$; існування нуля;
3. $\forall a \ \exists -a \quad a+(-a)=0$; існування оберненого елемента по додаванню;
4. $a+b=b+a$; комутативність додавання,

тобто поле – абелева група по додаванню;

5. $a(bc)=(ab)c$; асоціативність множення;
6. $\exists 1 \in M \quad a1=1a=a$; існування одиниці;
7. $\forall a \neq 0 \ \exists a^{-1} \quad a^{-1}a=1$; існування оберненого елемента по множенню;
8. $ab=ba$; комутативність множення,

тобто поле – абелева група по множенню;

9. $a(b+c)=(ab)+(ac)$ дистрибутивність множення відносно додавання.

Приклади.

1. Z – кільце цілих раціональних чисел, комутативне кільце з одиницею.
2. Q – кільце раціональних чисел, комутативне кільце з одиницею, поле.

3. R – кільце дійсних чисел, комутативне кільце з одиницею, поле.
4. C – кільце комплексних чисел, комутативне кільце з одиницею, поле.
5. $\langle Z,+,*\rangle$, де Z – множина всіх цілих чисел, – комутативне кільце з одиницею.
6. $\langle Z_n,+,*\rangle$, де Z – множина всіх цілих чисел, n – довільне натуральне число, – комутативне кільце з одиницею.

Якщо в кільці $\exists x \neq 0 \quad \exists y \neq 0 \quad x * y = 0$, то x називається *лівим*, а y – *правим дільником нуля.*

Комутативне кільце з одиницею, яке не має дільників нуля, називається *областю цілісності*.

Наприклад, цілі числа $\langle Z,+,*\rangle$ являються областю цілісності.

Приклади полів.

1. $\langle R,+,\cdot\rangle$, де R – множина всіх дійсних чисел, є полем дійсних чисел.
2. $\langle Q,+,\cdot\rangle$, де Q – множина всіх раціональних чисел, є полем раціональних чисел.

Приклади кілець.

1. $\langle Z,+,*\rangle$, де Z – множина всіх цілих чисел, – комутативне кільце з одиницею.
2. $\langle Z_n,+,*\rangle$, де Z – множина всіх цілих чисел, n – довільне натуральне число, – комутативне кільце з одиницею.

Завдання для індивідуальної роботи.

1. Довести, що якщо в напівтрупі існує правий одиничний елемент і для кожного елемента існує правий обернений, то ця напівгрупа являється групою.
2. На деякій множині чисел операція задана таким чином: $ab=b$. Довести, що ця множина є напівгрупою.
3. Довести, що в множині з завдання 2. існує лівий одиничний елемент.
4. Довести, що в множині з завдання 2. для кожного елемента існує правий обернений елемент.
5. Чи буде напівтрупа з операцією $ab=b$ групою?
6. Навести приклад скінченої напівтрупи з законом скорочення зліва, яка не є групою.
7. Знайти підгрупи групи цілих чисел по додаванню.
8. Знайти підгрупи групи раціональних чисел по додаванню.
9. Знайти підгрупи мультиплікативної групи дійсних чисел, відмінних від нуля.

10. Знайти елементи скінченого порядку в мультиплікативній групі дійсних чисел, відмінних від нуля.
11. Знайти елементи скінченого порядку в мультиплікативній групі комплексних чисел, відмінних від нуля.

Підкільце. Ідеал.

Нехай K – кільце.

Означення. Непуста підмножина $M \subset K$ називається *підкільцем кільця К,* якщо ця множина M є кільцем відносно тих дій, які введені в K.

Означення. Підкільце M кільця K називається *лівим ідеалом кільця К*, якщо

$$\forall r \in K \quad \forall m \in M \quad rm \in M .$$

Означення. Підкільце M кільця K називається *правим ідеалом кільця К*, якщо

$$\forall r \in K \quad \forall m \in M \quad mr \in M .$$

Означення. Лівий ідеал, який є і правим ідеалом, називається *двостороннім ідеалом* або просто *ідеалом.*

Приклади.

1. K – кільце, $M \subset K$, $M=\{0\}$. M – підкільце і називається *нульовим підкільцем. M* – лівий, правий і двосторонній ідеал: $0a = a0 = 0$.
2. $M=K$, M – підкільце, двосторонній ідеал.

Нехай K– кільце, M – підмножина.

Ознака підкільця I. Нехай $a,b \in M$. Для того, щоб M було підкільцем необхідно і достатньо, щоб:

1) $a+b \in M$;
2) $ab \in M$;
3) $0 \in M$;
4) $\forall a \ \ -a \in M$.

Ознака підкільця II. Підмножина M множини K є підкільцем K тоді і тільки тоді, коли $\forall a,b \in M$

1) $a-b \in M$;
2) $ab^{-1} \in M$.

Ознака ідеала. Підмножина $M \subset K$ є лівим ідеалом K тоді і тільки тоді, коли $\forall a,b \in M$

1) $a\text{-}b \in M$;

2) $\forall r \in K \ \forall a \in M \ ra \in M$.

Аналогічно для правого і двостороннього ідеалу.

Означення. Асоціативне кільце K з одиницею e називається *тілом* , якщо $\forall a \in K \ a \neq 0 \ \exists a^{-1} \in K : aa^{-1} = a^{-1}a = e$.

a^{-1} називається оберненим до a.

Комутативне тіло є *полем.*

Твердження. Нехай K – поле, M – підкільце K. M є полем тоді і тільки тоді, коли

$$\forall a \in M, a \neq 0 \ a^{-1} \in M .$$

Зауваження. Довільне поле K містить принаймні два елементи: 0 і 1, $0 \neq 1$.

Теорема. Будь-яке скінчене тіло є полем.

Приклади.

1. Множина раціональних чисел Q є полем, причому простим полем, тобто таким, що не містить нетривіального підполя.
2. Нехай K – тіло. Позначимо $(K)_n$ – множина всіх квадратних матриць порядка n з елементами, що належать тілу K. Відносно дій додавання і множення матриць $(K)_n$ буде кільцем. Одиницею цього кільця буде $e = \begin{pmatrix} e & & 0 \\ & \ldots & \\ 0 & & e \end{pmatrix}$.

   $(K)_n$ називається *повним матричним кільцем над тілом K.* Множення виконується так:

   нехай $A=(\alpha_{ij})$, $B=(\beta_{ij})$, тоді $C=AB$, $C=(\gamma_{ij})$,

   де $\gamma_{ij} = \sum_{j=1}^{n} \alpha_{ij}\beta_{ij}$, ($\alpha\beta \neq \beta\alpha$ в тілі K).

Література до курсу

Основна

1.Ілюшин В.І. Математична логіка та теорія алгоритмів. Саратов, 1986.
2.Калабеков Б.А.Цифрові пристрої та мікропроцесорні пристрої. Москва, 1987.
3.Чен Ч., Лі Р. Математична логіка та автоматичне доведення теорем. Наука, М., 1983.
4.Шестаков В.І. Математична логіка та автоматика, 1958, №6, 1959,№1.
5.Вітенько І.В. Математична логіка (курс лекцій). Ужгород, 1971.
6.Вітенько І.В. Конструктивні операції. Ужгород, 1972.
7.Витенько И.В. Схемы, алгоритмы и многообразия. Ужгород, 1970.
8.Новиков Ф.А. Дискретная математика для программистов. Санкт-Петербург, 2001.

Додаткова

1. Дж. Робинсон. Машинно-ориентированная логика, основанная на принципе резолюции. В кн. Кибернетический сборник, вып 7. Наукова думка, К., 1970.
2. С.К. Клини. Математическая логика. Мир, М., 1973.
3. Н.Нильсон. Искусственный интеллект. Мир, М., 1973.
4. Дж. Шенфилд. Математическая логика. Наука, М., 1975.
5. Э.Хант. Искусственный интеллект. Мир, М., 1978.
6. Г.М. Адельсон-Вельский и др. Машина играет в шахматы. Наука, М., 1983.
7. А. Эндрю. Искусственный интеллект. Мир, М., 1985.

Зміст