Національна академія наук України
Інститут кібернетики імені В. М. Глушкова

На правах рукопису

КРИВИЙ Сергій Лук'янович

УДК 51:681.3

ІТЕРАТИВНІ МЕТОДИ АНАЛІЗУ ПРОЦЕДУРНИХ ПРОГРАМ

01.05.01 — теоретичні основи інформатики та кібернетики

Автореферат дисертації на здобуття наукового ступеня доктора фізико-математичних наук

Київ 1997

007

ЛННБ України ім.В.Стефаника
00751769 (Z)

Дисертацією є рукопис.

Роботу виконано в Інституті кібернетики ім. В. М. Глушкова НАН України.

Науковий консультант: член-кореспондент НАН України, доктор фізико-математичних наук, професор ЛЕТИЧЕВСЬКИЙ О. А.

Офіційні опоненти: член-кореспондент НАН України, доктор фізико-математичних наук, професор, радник дирекції Інституту кібернетики ім. В. М. Глушкова НАН України ЮЩЕНКО К. Л.,

член-кореспондент НАН України, доктор фізико-математичних наук, професор, директор Інституту програмних систем НАН України АНДОН П. І.

доктор технічних наук, професор, декан факультету інформатики та обчислювальної техніки Національного технічного університету «Київський політехнічний інститут» ПАВЛОВ О. А.

Провідна організація: Київський національний університет імені Т. Г. Шевченка.

Захист відбудеться «26» грудня 1997 р. о 11 год. на засіданні спеціалізованої вченої ради Д 01.39.02 при Інституті кібернетики імені В. М. Глушкова НАН України за адресою:

252022 Київ 22, проспект Академіка Глушкова, 40.

З дисертацією можна ознайомитись у науково-технічному архіві інституту.

Автореферат розісланий «14» листопада 1997 р.

Учений секретар
спеціалізованої вченої ради СИНЯВСЬКИЙ В. Ф.

АВ-38.811

ЗАГАЛЬНА ХАРАКТЕРИСТИКА РОБОТИ

Актуальність проблеми. Методам автоматизації проектування, аналізу і обгрунтування програм на протязі всього розвитку програмування як науки приділялась велика увага дослідників в усьому світі. Виникнення цих методів тісно пов'язано з практичною діяльністю в програмуванні, і перш за все з такими складними проблемами, як верифікація і оптимізація програм. При вирішенні різноманітних проблем, пов'язаних з проектуванням якісного програмного забезпечення сучасних обчислювальних систем, ці проблеми відіграють першорядну роль.

Проблема (часткової) верифікації програми P полягає в тому, щоб за заданими висловлюваннями α і β про програму P визначити істинність висловлювання β на значеннях вихідних змінних програми P за умови, що значення її вхідних змінних задовольняють висловлюванню α.

Проблема оптимізації програм полягає в тому, щоб за заданою програмою побудувати еквівалентну їй другу програму, яка має кращі якісні характеристики (вищу швидкодію, менший об'єм пам'яті, кращу структурованість і т.д.).

Часткові розв'язки цих проблем (з причин їх алгоритмічної нерозв'язуваності в загальному випадку) досить складні і в кожному конкретному випадку вимагають детального аналізу програми, але вони можуть бути суттєво полегшені, якщо для заданого стану програми відомо достатньо багато її інваріантів, тобто таких тверджень, які виконуються всякий раз при проходженні процесу обчислень у програмі через цей стан. У деяких випадках такого роду інформацію можна збирати автоматично за допомогою методів, за якими закріпилась назва *методів потокового аналізу програм* і які є предметом дослідження в цій роботі.

Враховуючи основні вимоги до оптимізаційних перетворень програм та їх верифікації, в дисертаційній роботі ставиться і розв'язується проблема автоматичного пошуку і генерації інваріантних співвідношень для програм з одним рівнем пам'яті з урахуванням властивостей алгебри даних, над якою працює програма. Як зазначалося вище, розв'язок цієї проблеми полегшує процес верифікації і оптимізації програм, але разом з тим являє собою досить складний етап аналізу програми. Ця складність полягає в тому, що модель програми повинна бути досить конкретною і разом з тим до-

ЦНБ ім. В. Вернадського НАН України

сить абстрактною для того, щоб носити загальний характер. Крім того, модель повинна мати властивість незалежності від конкретних мов програмування і разом з тим бути застосовною до будь-якої процедурної мови програмування. Одержані алгоритми повинні бути ефективними, тобто генерувати по можливості за найкоротший проміжок часу інваріанти або, що те саме, виконувати ефективно етап аналізу програми і, що найголовніше, враховувати властивості алгебри даних, оскільки без урахування цих властивостей неможливі її глибокі оптимізаційні перетворення чи успішна верифікація.

Задача генерації інваріантних співвідношень програм з урахуванням особливостей алгебри даних уперше в явному вигляді була сформульована О.А.Летичевським у 1972 році на міжнародному симпозіумі, який проходив в Новосибірську. В наступному 1973 році Г. Кілдалл висловив аналогічну ідею і описав алгоритм генерації інваріантних співвідношень вигляду r - c, де r - змінна, а c - константа. Робота Кілдалла стимулювала інтенсивний розвиток досліджень методів потокового аналізу програм для вивчення і обгрунтування багатьох властивостей процедурних програм (задач оптимального розподілу регістрів, вилучення надлишкових обчислень, пов'язаних із спільними підвиразами, згортки констант, перевірки коректності типів даних, аналізу змінних на життєвість тощо). Цей період завершився розглядом результатів і підведенням підсумків у 1982 році міжнародною конференцією в Копенгагені. Наслідком підведення цих підсумків став вихід у світ тепер добре відомої книги "Program Flow Analysis: Theory and Applications". Наступний етап розвитку та застосувань алгоритмів потокового аналізу програм настав з появою та активним вивченням функціональних та логічних мов програмування. Відомі приклади, коли застосування методів потокового аналізу логічних програм до оптимізації таких програм дають прискорення в 40-50 разів. Підсумки розвитку методів потокового аналізу програм, або, як їх ще називають, методів статичного аналізу програм, за останні роки були підведені в вересні 1996 року на міжнародній конференції в Німеччині (м. Аахен).

З одного боку, аналізуючи стан справ, які склалися на даний час, потрібно зауважити, що проблема генерації програмних інваріантів ставиться і розв'язується багатьма дослідниками без врахування властивостей алгебри даних або з врахуванням надзвичайно простих властивостей, оскільки, як випливає з аналізу і приватних контактів з ученими як іноземних держав, так і з країн СНД, основною перепоною є складність проблеми. З іншого боку, давно і добре відомо, що найбільш суттєві чи то оптимізаційні перетворення, чи то перетворення з метою верифікації програм випливають саме з властивостей алгебри даних програми або, більш загально, з семантичних властивостей предметної області. В зв'язку з ситуацією, що складається, виникає необхідність досліджувати програми, розглядаючи їх разом з алгебрами даних, над якими вони працюють. Крім того, необхідно знайти розумні обмеження для того, щоб поставлені задачі мали алгоритмічне вирішення за сприйнятний час.

Отже, створення математичної теорії, узагальнюючих і водночас прагматичних методів автоматичного аналізу для програм з урахуванням властивостей предметної області та розробка ефективних алгоритмів такого аналізу є актуальною проблемою комп'ютерної науки.

Об'єктом дослідження в дисертаційній роботі є графова модель програми - так звані U-Y-схеми програм над пам'яттю - і математичні методи автоматичного аналізу таких моделей з різноманітними алгебрами даних.

Мета роботи. Метою досліджень є розробка загальної математичної теорії для аналізу широкого класу реальних програм і алгоритмів та побудова ефективних алгоритмів аналізу і генерації інваріантних співвідношень для таких програм з урахуванням властивостей їх алгебри даних.

Основні задачі дослідження. Враховуючи основні вимоги до сучасного програмного забезпечення і у відповідності з названою метою в дисертації ставляться і вирішуються такі задачі:

- розробка теоретичних основ і побудова ефективних алгоримів генерації інваріантних співвідношень для програм з різноманітними алгебрами даних, які забезпечують певну повноту одержаних множин інваріантів;

- розробка і дослідження методів та алгоритмів генерації інваріантних співвідношень для програм, алгебри даних яких не задовольняють критеріям повноти, з метою встановлення характеру цієї неповноти і пошуку шляхів підвищення генеративної спроможності одержаних алгоритмів;

- пошук і встановлення розумних обмежень, які накладаються на програми і їх алгебри даних, з метою окреслення рамок застосування розроблених методів та алгоритмів;

- застосування розроблених методів аналізу до оптимізації програм, програм з процедурами і до процесу проектування ефективних алгоритмів та програм;

- реалізація розроблених алгоритмів в мультипроцесорному макроконвейєрному комплексі ЄС 1766 з метою виконання оптимізаційних перетворень мультимодульних програм, а також для проведення експериментів на персональній ЕОМ в системі алгебраїчного програмування АПС, розроблених в Інституті кібернетики ім. В.М. Глушкова НАН України.

Методи досліджень. В основі методів досліджень лежать методи сучасної загальної алгебри, теорії автоматів, математичної логіки, теорії формальних мов, структур даних і теорії графів.

Наукова новизна результатів, що виносяться на захист полягає у такому:

- запропоновано та розроблено загальний алгеброавтоматний підхід до розв'язання проблеми автоматичної побудови множини інваріантних співвідношень для програм з урахуванням властивостей їх алгебри даних;

- запропоновано та досліджено два основних методи пошуку програмних інваріантів - метод верхньої і метод нижньої апроксимації, а також побудовані відповідні алгоритми для цих методів;

- досліджено властивості методів верхньої і нижньої апроксимації, а також відповідних їм алгоритмів, для випадку мови рівностей, як найбільш уживаної, сформульовані основні задачі теорії програмних інваріантів для цієї мови і встановлені умови, за виконання яких алгоритм верхньої апроксимації дає повні множини програмних інваріантів, а також показано, що у випадку порушення умов повноти алгоритм верхньої апроксимації дає максимально можливу множину програмних інваріантів;

- встановлено, що алгоритм верхньої апроксимації не є виродженим і дає повні множини програмних інваріантів для програм, алгебра даних яких є однією із таких алгебр: а) абсолютно вільна Ω-алгебра, б) комутативний Ω-групоїд (групоїд) з нулем і скороченнями, в) Ω-напівгрупа (напівгрупа) з умовою фінітарності, г) скінченномірний лінійний афінний простір, д) вільна комутативна напівгрупа, е) вільна абелева група;

- показано, що в загальному випадку алгоритм верхньої апроксимації дає хоча і неповну множину інваріантних співвідношень для програм, алгебра даних яких є вільною групою, але ця множина являє собою максимально можливу множину інваріантів при заданій апроксимації;

- встановлено, що алгоритм верхньої апроксимації для випадку мови рівностей і лінійних нерівностей у програмах, алгебра даних яких є скінченномірним лінійним афінним простором, за відповідних обмежень дає повну множину інваріантних співвідношень, тобто втрата співвідношень можлива лише завдяки обмеженням;

- на основі теорії програмних інваріантів запропонована методика оптимізаційних перетворень програм з процедурами, а також методика оптимізаційних перетворень при трансформаційному методі проектування ефективних програм і алгоритмів;

- розроблено серію ефективних алгоритмів обчислення конгруентних замикань виразів у скінченновизначених алгебрах, таких, як а) Ω-алгебра термів, б) комутативний Ω-групоїд, в) Ω-напівгрупа з умовою фінітарності; г) комутативна Ω-напівгрупа, д) абелева Ω-група, які складають основу алгоритмів аналізу програм;

- встановлено такі результати загальноматематичного характеру:

а) спадна (зростаюча) послідовність нормальних конгруенцій скінченнопородженої Ω-алгебри задовольняє умові обриву спадних (зростаючих) ланцюгів, а отже, і умовам максимальності і індуктивності;

б) перетин скінченнопороджених конгруенцій, що зберігають ациклічність термів, у випадку

- абсолютно вільної Ω-алгебри,
- комутативного Ω-групоїда (групоїда) з нулем і скороченнями,
- Ω-напівгрупи (напівгрупи) з умовою фінітарності і скороченнями є скінченнопородженою конгруенцією.

Таким чином, в дисертації висунута і розроблена теорія програмних інваріантів для програм з врахуванням особливостей алгебри даних; досліджено програми над цілим рядом класичних алгебр даних, які найбільш часто зустрічаються як в теорії, так і в практичних застосуваннях.

Практична цінність та реалізація. Загальна математична теорія аналізу потоків даних у програмі і наявність ефективних алгоритмів генерації інваріантних співвідношень відкриває нові напрямки і перспективи практичного застосування в області оптимізації, верифікації та обгрунтування програм, а також у методах проектування і розробки ефективних алгоритмів і програм.

Отримані в дисертації результати частково були реалізовані в математичному забезпеченні мультипроцесорного обчислювального комплексу ЄС 1766, в експериментальній системі алгебраїчного програмування АПС-1, розроблених в Інституті кібернетики ім. В.М. Глушкова НАН України, а також в учбовому процесі.

Дисертаційна робота виконана в рамках НДР по створенню мультипроцесорного обчислювального комплексу з макроконвейєрною обробкою даних ЄС 1766 (1982-1991 рр.): за проектом 857 "Розробка проекту багатопроцесорної ЕОМ з макроконвейєрною організацією обчислень"; у рамках Державної науково-технічної програми 6.3.1, тема В.Г.Е.100.09 "Розробити ефективні методи доведення тверджень і розв'язання задач на алгебрологічних моделях предметних областей"; а також за науково-дослідними проектами програми фундаментальних досліджень ДКНТ України 1995-1997 рр. "Розробка методів та засобів розв'язування логічних задач на алгебрологічних моделях предметних областей" (номер теми И.П.100.01), проекту INTAS-93-1702 "Symbolic Computing" у рамках Європейської наукової програми INTAS, 1994-1996 рр. та українсько-американського проекту. SRDF "Ефективні символьні обчислення, що базуються на логіці переписування та алгебраїчному програмуванні" (1997-1999).

Апробація роботи. Результати, наведені в дисертаційній роботі, доповідалися практично на всіх всесоюзних та республіканських конференціях з теоретичного і прикладного програмування, прикладної математичної логіки і алгебри, а також у багатьох університетах на семінарах. Зокрема, на міжнародних і всесоюзних конференціях: 1983 р. "Системное и теоретическое программирование" (м. Кишинів), 1983 р. "Прикладная алгебра" (м. Новосибірськ), 1981 і 1986 роки "Проблемы совершенствования, синтеза, тестирования, верификации и отладки программ" (м. Рига), 1985 і 1991 роки "Смешанные вычисления и оптимизация программ" та 1988 р. "Методы трансляции и конструирования программ" (м. Новосибірськ), 1986 р. "Теоретическое и прикладное программирование" (м. Таллінн), 1988 р. "Проблемы создания супер-ЭВМ, супер-систем и их применение" (м. Мінськ), 1990 і 1995 роки 2-га і 4-та Міжнародні конференції "Прикладная логика" (м. Іркутськ), 1989 р. 2-га Всесоюзна школа-семінар з алгебри і логіки (м. Орджонікідзе), 1989, 1990, 1991 роки "Высокопроизводительные машины и системы" (м. Коктебель, м. Уфа, м. Коктебель), 1990 рік Міжнародна конференція з алгебри і логіки (присвячена пам'яті А.Д. Тайманова, м. Алма-Ата), в університетах м. Киева, Ужгорода, Риги, Санкт-Петербурга, Новосибірська, Іркутська, Роттердама (Голландія), Уппсали (Швеція).

Публікації. Основні результати дисертаційної роботи повно викладено у 41 друкованій праці, з них 17 найважливіших включають одну колективну монографію і один учбовий посібник для студентів технічних вузів, а також статті у провідних вітчизняних фахових журналах "Кибернетика и системный анализ", "Доповіді НАН України", у працях Міжнародних та Всесоюзних конференцій з прикладної математичної логіки, загальної алгебри, системного та теоретичного програмування останніх років. З цих 17 робіт, вісім опубліковано без співавторів.

Структура і обсяг дисертації. Дисертація складається із вступу, семи розділів, переліку основних результатів і висновку, списку літератури і вміщує 254 сторінки основного тексту та 22 малюнки. Список літератури налічує 143 найменування. Загальний обсяг роботи 266 сторінок.

ЗМІСТ РОБОТИ

Перший розділ роботи складають основні поняття загальної алгебри, теорії скінченних автоматів, теорії графів і графових структур даних, у термінах яких іде викладення основного матеріалу дисертації. До таких понять, зокрема, відносяться поняття вільної універсальної алгебри заданого класу алгебр, повної гратки, поняття мови предикатів першого порядку, скінченного автомата без виходів та регулярної мови, графа, орграфа і складеного об'єкта як графової структури даних.

У другому розділі дисертації вводиться поняття конгруентного замикання виразу в скінченновизначеній алгебрі заданого класу алгебр, відносно множини тотожних і визначальних співвідношень, розробляються ефективні алгоритми обчислення конгруентних замикань для цих класів алгебр. Необхідність в цих алгоритмах пов'язана з тим, що вони складають основу алгоритмів аналізу U-Y-програм, про які йде мова в наступних розділах.

Нехай $T(\Omega,X)$ означає Ω-алгебру термів над алфавітом X сигнатури Ω, яка визначається множиною тотожних співвідношень E, і нехай $\theta = \{t_1 = t_1',\dots,t_n = t_n',\dots\}$ - деяка множина рівностей термів, де $t_i,t_i' \in T(\Omega,X)$, $i = 1,\dots,n,\dots$. Кожну рівність із θ можна розглядати як пару (t_i,t_i'), а саму множину θ - як бінарне відношення на $T(\Omega,X)$.

Алгебраїчним замиканням множини θ відносно множини тотожних співвідношень E називається найменша множина $C(\theta)$, яка включає рефлексивне, симетричне і транзитивне замикання множини θ, усі тотожності із E і для будь-якої операції ω із Ω арності n разом з парами $(t_1,t_1'),\dots,(t_n,t_n')$ включає пару $(\omega(t_1,\dots,t_n), \omega(t_1',\dots,t_n'))$. Таку пару будемо називати елементом алгебраїчного замикання. Множина θ називається *алгебраїчно замкнутою*, якщо $C(\theta) = \theta$. Підмножина P алгебраїчно замкнутої множини θ називається *алгебраїчним базисом* множини θ, якщо $C(P) = \theta$. Безпосередньо з наведених визначень випливає, що $C(\theta) = \theta$ тоді і тільки тоді, коли θ - конгруенція на $T(\Omega,X)$.

Для успішного розв'язання задач теорії програмних інваріантів розробляється загальний алгоритм обчислення конгруентних замикань виразів для таких скінченновизначених алгебр, як абсолютно вільні Ω-алгебри, комутативні Ω-групоїди, комутативні Ω-напівгрупи, абелеві Ω-групи, Ω-напівгрупи з умовою фінітарності. Досліджуються властивості загального алгоритму при конкретизації його до кожної окремо взятої із цих алгебр. Структурно, розділ 2 складається із п'яти параграфів. Зокрема, у першому параграфі розглядається загальний алгоритм обчислення конгруентного замикання. У другому параграфі досліджуються властивості загального алгоритму для таких алгебр, як абсолютно вільні Ω-алгебри і комутативні Ω-групоїди. Показано, що для цих алгебр задачу обчислення конгруентного замикання можна сформулювати у більш загальному вигляді - у вигляді задачі обчислення конгруентного і симетричного конгруентного замикань для скінченних автоматів без виходів. Властивості загального алгоритму стосовно вказаних алгебр характеризуються такими твердженнями:

Теорема 2.2.1. Часова складність алгоритму обчислення конгруентного замикання терма t в абсолютно вільній алгебрі пропорціональна величині $O(\log(k)\cdot m\cdot\log(\log(k)\cdot m))$, де m - число переходів в автоматі A, яким представляється цей терм, а k - максимальна арність операцій із сигнатури Ω.

Теорема 2.2.2. Часова складність алгоритму обчислення конгруентного симетричного замикання терма t в комутативному групоїді пропорціональна величині $O(km \log m)$, де m - число переходів в автоматі A, яким представляється цей терм, а k - максимальна арність комутативної операції.

Теорема 2.4.3. Часова складність алгоритму обчислення конгруентного замикання терма t в Ω-напівгрупі з умовою фінітарності є поліноміальною ($O(N^4)$).

Теорема 2.2.5. Алгоритм обчислення конгруентного замикання терма t в a) комутативній Ω-напівгрупі є експоненціальним; b) абелевій Ω-групі - поліноміальним.

Завершується розділ оглядом деяких застосувань побудованих алгоритмів до розв'язування проблеми загальних підвиразів і ідентифікації зразків у термах.

У третьому розділі в першому параграфі вводяться графова модель програми - U-Y-схема програми над пам'яттю і деякі характеристики цієї моделі.

У другому параграфі визначається основне поняття дисертаційної роботи - поняття програмного інваріанта в стані U-Y-програми.

Нехай A - U-Y-схема програми над пам'яттю R, яка інтерпретована на області даних D (U-Y-програма), П - сигнатура елементарних умов а Ω - сигнатура операцій цієї програми і L - мова, в якій записуються твердження про властивості інформаційного середовища B. Відносно мови L будемо припускати, що будь-яке її речення (яке називається також умовою) може бути виражено формулою $\Phi(r)$ мови числення предикатів першого порядку, яка включає вільні змінні із кортежа $r = \langle r_1,\dots,r_m\rangle$ і яка інтерпретована на області даних D. Сигнатура цього числення включає всі символи сигнатур Ω і П. Нехай u(r) - деяка умова із L.

Визначення. Умова $\Phi(r)$ із L називається *інваріантом стану a* U-Y-програми A *відносно умови* u(r), якщо вона істинна при кожному проходженні стану a в процесі виконання програми A для тих початкових станів пам'яті із B, на яких істинна умова u(r). Умова u(r) називається *початковою.* Якщо вона тотожно істинна на D, то $\Phi(r)$ будемо називати просто *інваріантом стану a*.

Дається загальна неформальна характеристика поняття інваріанта і формулюються два методи генерації програмних інваріантів, які називаються методом верхньої і методом нижньої апроксимації і які грунтуються на наведених нижче формулах (3.2.1) і (3.2.2).

Число різних можливих шляхів у програмі (за наявності в ній хоча б одного циклу) може бути нескінченним, і тоді процес побудови умови стану a теж може стати нескінченним. Але нехай $a_1,\dots,a_k$ - всі стани U-Y-програми A, які зв'язані переходами (a_i,u_i,y_i,a) зі станом a і N_i - множина інваріантів стану a_i відносно деякої початкової умови і $ef(N_i,u_i,y)$ - функція генерації співвідношень для оператора y з урахуванням множини співвідношень

N_i і'умови u_i. Тоді, очевидно, що множина

$$\bigcap_{i=1}^{k} ef(N_i, u_i, y_i)$$

буде інваріантом стану a відносно тієї ж початкової умови. Цей простий факт є відправною точкою для побудови двох ітеративних методів генерації інваріантів.

У першому з них, що називається *методом нижньої апроксимації (МНА)*, ітеративний процес задається рекурентним співвідношенням

$$N_a^{(n)} = \bigcap_{(a',u,y,a)\in S} ef(N_{a'}^{(n-1)},u,y),\ n > 0,\ a,a' \in A, \quad (3.2.1)$$

а початкове наближення $\{N_{a_0}^{(0)}\}$ - рівностями $N_{a_0}^{(0)} = \{u\}$ і $N_a^{(0)} = \varnothing$ для $a \neq a_0$. З властивості монотонності функції ef випливає, що $N_a^{(0)} \subset N_a^{(1)} \subset \ldots$ незалежно від стану a. Вказаний ітеративний процес може завершитися через скінченне число кроків у результаті стабілізації послідовностей $N_a^{(i)}$ для всіх $a \in A$ або може тривати нескінченно, але перевага такого методу в тому, що, не чекаючи стабілізації процесу обчислень, їх можна перервати, оскільки будь-яка множина $N_a^{(n)}$ включається в множину інваріантів стану a.

У другому методі, який називається *методом верхньої апроксимації (МВА)*, ітеративний процес задається рекурентним співвідношенням

$$N_a^{(n)} = N_a^{(n-1)} \cap \left(\bigcap_{(a',u,y,a)\in S} ef(N_{a'}^{(n-1)},u,y)\right),\ n>0,\ a,a' \in A, \quad (3.2.2)$$

а початкове наближення визначається рівністю $N_{a_0}^{(0)} = \{u\}$ і деякою сукупністю простих шляхів, що покривають всю множину станів U-Y-програми A. Обчислення початкового наближення виконується вздовж цих шляхів, починаючи в $N_{a_0}^{(0)}$: якщо для деякого $a \in A$ уже відомо $N_a^{(0)}$, перехід (a,u,y,a') належить одному з шляхів заданої сукупності, а $N_{a'}^{(0)}$ ще не відомо, то покладаємо $N_{a'}^{(0)} = ef(N_a^{(0)},u,y)$. Із співвідношення (3.2.2) видно, що для будь-якого $a \in A$ мають місце включення $N_a^{(0)} \supset N_a^{(1)} \supset \ldots$ і, отже, шукана сукупність інваріантів може бути одержана тільки після стабілізації ітеративного процесу. Оскільки процес пошуку інваріантів може бути нескінченним, то цей факт є недоліком методу МВА, але у випадку результативного завершення він генерує більш повні системи інваріантів, ніж метод МНА.

Аналіз методів генерації інваріантів показує, що скінченність процесу пошуку інваріантів залежить від виразності мови умов L. Найбільш уживаними мовами умов є мови типу рівностей і нерівностей, оскільки практично кожна мова програмування включає предикати рівності і нерівності. Розробка методів пошуку інваріантів уже для цих мов пов'язана із значними труднощами. Дійсно, формули (3.2.1) і (3.2.2) вимагають розв'язання, і по можливості ефективного, таких задач, як обчислення значення функції ef, обчислення перетину двох множин умов, стабілізації процесу генерації. У зв'язку з цим мова інваріантів звужується до мови рівностей і формулюються основні задачі теорії програмних інваріантів для цієї мови: задача про співвідношення, задача про перетин і задача про стабілізацію. Вводиться поняття нормальної конгруенції і доводяться твердження про нормальні конгруенції і функцію генерації співвідношень ef, які наведені нижче.

Нехай $T_D(R)$ - вільна алгебра класу, якому належить алгебра даних (D,Ω), M - алгебраїчно замкнута множина рівностей на $T_D(R)$. Оскільки M - конгруенція, то відповідну фактор-алгебру будемо позначати через $T_D(R)/M$, її елементи через $t(\mathrm{mod}\ M)$, $t \in T_D(R)$, а рівність термів - $t = t'(\mathrm{mod}\ M)$. З кожним оператором присвоювання $y = (r_1:=t(r),\ldots,r_m:=t_m(r))$ ($t_i \in T_D(R)$) і алгебраїчно замкнутою множиною M зв'яжемо гомоморфізм

$$h_y: T_D(R) \to T_D(R)/M,$$

покладаючи $h_y(r_i) = t_i(\mathrm{mod}\ M)$.

Конгруенція M на $T_D(R)$ називається *нормальною*, якщо вона є ядром ендоморфізму алгебри $T_D(R)$, тобто $T_D(R)/M$ ізоморфна підалгебрі алгебри $T_D(R)$.

Нехай $ef(M,y)$ (звуження функції $ef(M,u,y)$) означає множину рівностей вигляду $t(r) = t'(r)$, таких, що $t(t_1,\ldots,t_m) = t'(t_1,\ldots,t_m) \in M$.

Лема 3.3.1. $ef(ef(M,y),y') = ef(M,yy')$.

Теорема 3.3.2. Якщо M - нормальна конгруенція, то $ef(M,y)$ - теж нормальна конгруенція.

Функція ef називається *дистрибутивною*, якщо для будь-яких M і M' справедливо $ef(M \cap M',y) = ef(M,y) \cap ef(M',y)$.

Теорема 3.3.3. Якщо множини M і M' алгебраїчно замкнуті, то функція ef дистрибутивна.

У третьому параграфі розробляються і досліджуються загальні властивості алгоритмів пошуку інваріантних співвідношень для мови рівностей і встановлюються такі властивості.

Нехай алгоритм МВА будує спадну послідовність множин співвідношень

$$N_1 \supset N_2 \supset N_3 \supset \ldots \supset N_n \supset \ldots, \qquad (3.3.1)$$

що має місце в деякому стані a U-Y-програми А.

Теорема 3.3.4. Якщо вільна алгебра $T_D(R)$ скінченнопороджена і кожний елемент N_i послідовності (3.3.1) є нормальною конгруенцією, то послідовність (3.3.1) стабілізується через скінченне число кроків.

Доведення опирається на такі твердження.

Лема 3.3.2. Якщо N і N' - деякі нормальні конгруенції на $T_D(R)$ і $N \subseteq N'$, то існують алгебри А і В, такі, що $A \subseteq B$ і алгебра В ізоморфна алгебрі $T_D(R)/N$, а алгебра А ізоморфна алгебрі $T_D(R)/N'$.

Лема 3.3.3. Якщо всі елементи N_i послідовності (3.3.1) нормальні конгруенції, то сукупність всіх цих елементів, поповнена, можливо, одним елементом, є повними гратками.

Позначимо повні гратки, про які йшла мова в наведеній вище лемі, через L і розглянемо сукупність L' фактор-алгебр

$$(T_D(R)/N_1)^1,\ (T_D(R)/N_2)^1, \ldots,\ (T_D(R)/N_n)^1, \ldots,$$

де N_1, $N_2, \ldots$, $N_n, \ldots$ - елементи послідовності (3.3.1), а $(T_D(R)/N_i)^1$ алгебри, що ізоморфні відповідно фактор-алгебрам $T_D(R)/N_i$ $(i = 1,2,\ldots)$, які існують в силу того, що N_i - конгруенції. Крім того, в силу леми 3.3.2 можна вважати, що алгебри сукупності L' упорядковані відношенням включення

$$(T_D(R)/N_1)^1 \subseteq (T_D(R)/N_2)^1 \subseteq \ldots \subseteq (T_D(R)/N_n)^1 \subseteq \ldots .$$

Лема 3.3.4. Відображення $f\colon L \to L'$, де $f(N_i) = (T_D(R)/N_i)^1$, є інверсним ізоморфізмом повних граток і

$$T_D(R)/N = f(N) = f(\bigcap_i N_i) = \bigcup_i f(N_i) = \bigcup_i (T_D(R)/N_i)^1,$$

тобто алгебри $T_D(R)/N$ і $\bigcup_i (T_D(R)/N_i)^1$ - ізоморфні.

Лема 3.3.5. Якщо $H = \bigcup_j (T_D(R)/N_j)^1$, то через скінченне число кроків послідовність

$$(T_D(R)/N_1)^1 \subseteq (T_D(R)/N_2)^1 \subseteq \ldots \subseteq (T_D(R)/N_n)^1 \subseteq \ldots \qquad (3.3.2)$$

стабілізується.

Рівність $(T_D(R)/N_n)^1 = (T_D(R)/N_{n+1})^1$ слід розуміти з точністю до ізоморфізму.

Лема 3.3.6. Якщо N_i, N_j - нормальні конгруенції, $N_i \subseteq N_j$ і $T_D(R)/N_i$ ізоморфна $T_D(R)/N_j$, то $N_i = N_j$.

З теореми 3.3.4 випливають такі наслідки.

Наслідок 3.3.1. Якщо для елементів послідовності (3.3.1) існують алгебраїчні базиси і, зокрема, скінченні алгебраїчні базиси, то ця послідовність стабілізується через скінченне число кроків.

Наведемо ще деякі властивості алгоритмів генерації інваріантних співвідношень, які випливають з теореми 3.3.4.

Теорема 3.3.5. Якщо в U-Y-програмі множина N_0 і перетин нормальних конгруенцій є нормальними конгруенціями, то алгоритм МВА (МНА) завершує свою роботу після скінченного числа кроків.

Теорема 3.3.4 дає теоретичну основу скінченності числа кроків алгоритмів генерації множин співвідношень, але в цій теоремі присутній деякий елемент неконструктивності - невідома довжина послідовності (3.3.1). При розгляді конкретних алгебр ця довжина може уточнятися. Але, не зважаючи на це, практична цінність цієї теореми полягає в тому, що вона показує, що повнота множин інваріантів залежить від точності алгоритмів розв'язку задач про співвідношення і про перетин. Якщо відповідні алгоритми дають точні розв'язки цих задач, то алгоритм МВА дає максимально можливі множини інваріантів для даної мови L.

Більш точні умови, при виконанні яких алгоритми генерації інваріантних співвідношень будують максимально можливі системи таких співвідношень, дає

Теорема 3.3.6. Якщо множина N_0 і множини співвідношень, що будуються в кожному стані U-Y-програми в процесі роботи алгоритму МВА, нормальні конгруенції і функція ef - дистрибутивна, то результат работи алгоритму МВА не залежить від порядку обходу станів U-Y-програми (і від вибору початкової сукупності простих шляхів), а множина інваріантів N_a для будь-якого стану a є A збігається з множиною

$$\bigcap_{l \in l(a_0,a)} ef(N_0,y_l).$$

Доведення теореми грунтується на таких лемах.

Лема 3.3.7. Якщо функція ef дистрибутивна і алгоритм МВА завершує свою роботу після скінченного числа кроків, то

$$(\forall a \in A)(N_a - \bigcap_{(a',u,y,a) \in S} ef(N_{a'},y).$$

Лема 3.3.8. Якщо функція ef дистрибутивна і алгоритм МВА завершує свою роботу після скінченного числа кроків, то

$$(\forall a \in A)(N_a = \bigcap_{l-l(a_0,a)} ef(N_0,y_1)).$$

Надалі, якщо множина інваріантів будь-якого стану a U-Y-програми збігається з множиною

$$\bigcap_{l-l(a_0,a)} ef(N_0,y_1),$$

то вона буде називатися *повною відносно* генератора інваріантів Gen, мови інваріантів L і початкової множини співвідношень N_0.

Теорема 3.3.6 дає умови, за яких алгоритм МВА генерує повні множини співвідношень відносно початкової сукупності співвідношень. Якщо ж функція ef не є дистрибутивною, то одержання повної системи співвідношень стає алгоритмічно нерозв'язуваною проблемою. Це випливає з такого твердження.

Теорема 3.3.7. Якщо функція ef монотонна, то можна вказати U-Y-програму A з початковою множиною співвідношень N_0, для якої не існує алгоритму, що генерує повну систему інваріантів відносно N_0 для цієї U-Y-програми.

У зв'язку з цією теоремою природно постає питання: а на яку множину інваріантів можна розраховувати у випадку недистрибутивності функції ef і які властивості цієї множини? Відповідь на це питання дає

Теорема 3.3.8. Якщо алгоритм МВА завершує свою работу після скінченного числа кроків і функція ef є монотонною, то одержана в результаті множина інваріантів N_a для будь-якого стану $a \in A$ є максимальною нерухомою точкою розв'язку системи рівнянь

$$\begin{cases} X_a = X_a \bigcap\limits_{(a',u,y,a) \in S} ef(X_{a'},y) \\ X_{a0} = N_0. \end{cases}$$

Наслідок 3.3.3. Результат работи алгоритму МВА не залежить від порядку вибору елементів із множини станів.

У четвертому розділі розглядаються U-Y-програми з такими алгебрами даних, для яких функція ef є дистрибутивною.

В першому параграфі розглядаються U-Y-програми з абсолютно вільною алгеброю даних. Має місце така

Теорема 4.1.11. Для будь-якої U-Y-програми алгоритм МВА (МНА) завершує свою роботу через скінченне числа кроків. Алгоритм МВА будує повну систему інваріантів і число кроків алгоритму обмежено величиною $const\cdot(m\cdot n)^2$, де $m = |R|$, R - множина змінних

U-Y-програми, n - число вершин у графі U-Y-програми.

У другому параграфі досліджуються U-Y-програми, алгебра даних яких є комутативним Ω-групоїдом з нулем і скороченнями відносно комутативної операції. Тут одержані такі результати.

Теорема 4.2.6. Для будь-якої U-Y-програми, алгебра даних якої є комутативним Ω-групоїдом з нулем та оберненими елементами відносно комутативної операції і початкова множина співвідношень N_0 є нормальною конгруенцією, алгоритм МЗА завершує свою роботу після скінченного числа кроків і будує повну сукупність інваріантів.

При розробці алгоритмів генерації інваріантів для програм над такою алгеброю даних був розроблений алгоритм розв'язку задачі про перетин **БАЗИС**, який будує алгебраїчний базис P перетину двох алгебраїчно замкнутих множин за їх скінченними алгебраїчними базисами P_1 і P_2 за умови *збереження ациклічності термів*. Ця умова формулюється так: терми, що входять в P_1 і P_2, є ациклічними і якщо співвідношення d є наслідком d' в P_1 чи P_2, то ліва і права частини d' - ациклічні терми.

Для алгоритму БАЗИС мають місце такі твердження.

Теорема 4.2.2. $C(P) = C(P_1) \cap C(P_2)$, де P_1 і P_2 - скінченні алгебраїчні базиси.

Наслідок 4.2.1. Якщо алгебра даних $T_D(R)$ є абсолютно вільною, то алгоритм БАЗИС будує множину P, таку, що $C(P) = C(P_1) \cap C(P_2)$.

Може скластися така ситуація, коли в базисі P_2 (P_1) можуть існувати елементи $d_1, d_2, \ldots, d_k$ ($k > 2$), такі, що, наприклад, $L(d_1) = L(d_2)$, $R(d_2) = L(d_3), \ldots, L(d_{k-1}) = L(d_k)$, $R(d_k) = R(d_1)$ є наслідками елементів базису P_1 (P_2), де $L(d)$ і $R(d)$ відповідно ліва і права частини елемента d. Тоді якщо $k < ar(\omega)$, то будуємо елемент вигляду $\omega(L(d_1),R(d_2),\ldots) = \omega(R(d_1),L(d_2),\ldots)$. Цей елемент будемо називати *елементом комутативного замикання* множини P_2 (P_1) в P_1 (P_2) *довжини k*.

Теорема 4.2.3. Базис P, побудований в результаті роботи алгоритму БАЗИС за скінченними базисами P_1 і P_2, є нескінченним тоді і тільки тоді, коли для комутативної операції ω із Ω - арності n існує хоча б один елемент комутативного замикання довжини строго меншої, ніж n.

Наслідок 4.2.2. Якщо всі комутативні операції із Ω є бінарними, то базис P скінченний.

Наслідок 4.2.3. Якщо алгебра $T_D(R)$ абсолютно вільна, то базис P скінченний.

Наслідок 4.2.4. Перетин скінченного числа скінченнопороджених конгруенцій, які задовольняють умові ациклічності термів, у випадку а) комутативного Ω-групоїда, б) комутативного групоїда, в) абсолютно вільної алгебри є скінченнопородженою конгруенцією.

У третьому параграфі розглядаються U-Y-програми, алгебри даних яких є фінітарними Ω-напівгрупами. Множина визначальних співвідношень P називається *фінітарною*, якщо множина всіх наслідків, одержаних у результаті виконання підстановок, для цієї множини скінченна. Скінченновизначена напівгрупа $T(\Omega,X)$ називається фінітарною, якщо множина її визначальних співвідношень P фінітарна.

Основні результати цього параграфа зводяться до таких.

Теорема 4.3.1. Множина $ef(\varnothing,y)$ має скінченний алгебраїчний фінітарний базис P.

Теорема 4.3.2. Якщо P - фінітарна множина, то $ef(P,y)$ - теж фінітарна множина.

Необхідно зауважити, що в множині співвідношень $ef(M,y)$ має місце закон скорочення.

Систему слів $S = \{q_1,q_2,\ldots,q_l\}$ вільної напівгрупи слів $F(R)$ над алфавітом R будемо називати *незалежною* або *системою*, що задовольняє *умові ізольованості початків (кінців)*, якщо жодне із слів q_j не можна представити у вигляді добутку $q_j = q_i q$ (або $q_j = q q_i$), де q_i, $q_j \in S$, а q - підходяще, можливо, пусте, слово, $i \neq j$, $i,j = 1,2,\ldots,l$.

Нехай $P = \{p_1,p_2,\ldots,p_k\}$ - скінченна множина слів із напівгрупи $F(R)$, а $L(P)$ означає піднапівгрупу напівгрупи $F(R)$, яка породжена словами із P.

Теорема 4.3.3. Множина $L(P)$ є регулярною множиною, тобто множиною, що зображається в деякому скінченному автоматі A. Якщо множина P задовольняє умові ізольованості початків, то автомат A має злиттями лише ті стани, які безпосередньо досяжні з початкового стану і заключних станів, тобто стани-злиття є стани вигляду $\delta(a_0,x) = \delta(a,x)$, де δ - функція переходів автомата, $a \in F$ - заключний стан, a_0 - початковий стан, а x пробігає всі початкові символи слів із P. Крім того, $\delta(a,x) \neq a_0$ для всіх $a \in A$ і $x \in R$.

Теорема 4.3.4. Перетин двох скінченнопороджених піднапівгруп вільної Ω-напівгрупи, множини твірних яких задовольняють умові ізольованості початків, є скінченнопородженою піднапівгрупою вільної Ω-напівгрупи.

Наслідок 4.3.1. Перетин скінченного числа скінченнопороджених піднапівгруп вільної напівгрупи, множини твірних яких задовольняють умові ізольованості початків, є скінченнопородженою піднапівгрупою вільної групи.

Наслідок 4.3.2. Напівгрупа, одержана в результаті перетину двох скінченнопороджених піднапівгруп вільної напівгрупи, множини твірних яких задовольняють умові ізольованості початків, породжується не більш ніж $m \cdot n$ числом твірних, де m і n - число твірних вихідних напівгруп.

Нехай $P_1 = \{p_1 = p_1',\ldots,p_n = p_n'\}$ і $P_2 = \{q_1 = q_1',\ldots,q_m = q_m'\}$ - два скінченні алгебраїчні базиси множин співвідношень, заданих на напівгрупі $T(\Omega,R)$ і $T(\Omega,R)$ фінітарна як щодо P_1, так і щодо P_2. Розглянемо транзитивне замикання $T(P_1)$ і $T(P_2)$--відношення безпосереднього слідування для множин P_1 і P_2, відповідно. В силу фінітарності напівгрупи $T(\Omega,R)$ множини $T(P_1)$ і $T(P_2)$ скінченні. Крім того, представлення $(T(\Omega,R,P_1)$ і $(T(\Omega,R,T(P_1))$ еквівалентні.

Нехай $a_1a_2\ldots a_k = b_1b_2\ldots b_s$ - деяке співвідношення із $T(P_1)$ $(T(P_2))$. Побудуємо множини Q_1 і Q_2, елементами яких є слова вигляду $x_{i1}{}^1x_{i2}{}^1\ldots x_{ik}{}^1$, що відповідають i-й рівності із $T(P_1)$, $i = 1,2$, де $x_{ij} = (a_j,b_j)$, a_j, b_j - або обидва непусті символи із R, або один з них пустий (одній і тій же рівності відповідає пара слів через симетричність відношення рівності). При цьому, якщо в одній із множин Q_1 або Q_2 серед символів x є такий, що відповідає діагональному елементу, то цей символ додається як нове слово в обидві множини, тобто додається елемент вигляду $x = (a,a)$, де $a \in R$. У силу закону скорочення множини Q_1 і Q_2 задовольняють умові ізольованості початків.

Побудуємо тепер множину твірних напівгрупи $L(Q_1) \cap L(Q_2)$, яка існує за теоремою 4.3.4 і є скінченною. Нехай $Q = \{p_1,\ldots,p_k\}$ - ця множина твірних. Елементам цієї множини відповідають рівності $q_i = q_i'$, $i = 1,2,\ldots,k$. Вилучимо із цієї множини рівностей елементи, які відповідають діагональним елементам, і рівності, що є наслідками решти. Нехай $P = \{t_{11} = t_{11}',\ldots,t_{s1} = t_{s1}'\}$ - множина побудованих таким чином рівностей.

Теорема 4.3.5. $C(P) = C(P_1) \cap C(P_2)$.

Наслідок 4.3.3. Перетин скінченного числа скінченнопороджених конгруенцій в а) фінітарній Ω-напівгрупі із скороченнями, б) фінітарній напівгрупі із скороченнями є скінченнопородженою конгруенцією.

Завершується параграф таким твердженням, яке підводить підсумки його змісту.

Теорема 4.3.8. Якщо алгебра даних $T_D(R)$ є вільною Ω-напівгрупою або вільною напівгрупою, то для будь-якої U-Y-програми над такою алгеброю даних з фінітарною множиною початкових співвідношень N_0 алгоритм МВА завершує свою роботу через скінченне число кроків і будує повну систему інваріантних співвідношень.

У четвертому параграфі розглядаються U-Y-програми, алгебри даних яких є скінченномірні лінійні афінні простори, абелеві групи і комутативні напівгрупи.

Основними результатами цього параграфа є такі.

Теорема 4.4.1. Для будь-якої U-Y-програми, алгебра даних якої є лінійний афінний простір, алгоритм МВА (МНА) завершує свою роботу через скінченне число кроків. Алгоритм МВА будує повну множину інваріантів і його часова складність пропорціональна величині $n \cdot m^{3,46}$.

Якщо $T_D(R)$ - вільна абелева група, то тоді її можна розглядати як скінченнопороджений модуль над кільцем цілих чисел Z. В цьому випадку елементи ef(M,y) мають груповий вигляд

$$\gamma_1 r_1' + \gamma_2 r_2' + \ldots + \gamma_m r_m' = 0,$$

де $\gamma_i \in Z$, $r_i' = r_i - c_i$, $r_i \in R$, c_i - константа, $i = 1,2,\ldots,m$. Можливість зведення задачі пошуку інваріантів програм над вільними абелевими групами даних до задачі пошуку таких у випадку, коли $T_D(R)$ - лінійний афінний простір, дає той факт, що будь-яка область цілісності ізоморфно вкладається в поле своїх дробів.

Кільце цілих чисел Z є областю цілісності, а полем його дробів - поле раціональних чисел. Отже, кільце Z можна ізоморфно вкласти в поле раціональних чисел і генерувати інваріанти, вважаючи, що $T_D(R)$ - лінійний афінний простір над полем раціональних чисел. Побудувавши інваріанти, їх необхідно привести до групового вигляду.

Якщо $T_D(R)$ - вільна комутативна напівгрупа, то елементи ef(M,y) мають напівгруповий вигляд

$$\gamma_1 r_1' + \ldots + \gamma_k r_k' = \gamma_{k+1} r_{k+1}' + \ldots + \gamma_m r_m',$$

де γ_i - цілі натуральні числа, $r_i' = r_i - c_i$, $r_i \in R$, c_i - константа. Користуючись прийомом вкладення однієї алгебраїчної структури в іншу, можна звести задачу пошуку інваріантів для програм над вільними комутативними напівгрупами даних до пошуку таких, коли $T_D(R)$ - вільча абелева група. Підставою для цього є те, що будь-яка комутативна напівгруппа із скорочнями ізоморфно вкладається в абелеву групу.

Оскільки закон скорочення в $T_D(R)$ має місце, то можна будувати інваріанти описаним вище способом для програм над лінійним афінним простором, а потім одержані інваріанти необхідно привести до напівгрупового вигляду. Таким чином, справедлива така теорема.

Теорема 4.4.4. Для будь-якої U-Y-програми, алгебра даних якої є а) лінійний афінний простір; б) вільна абелева група; в) вільна комутативна напівгрупа, алгоритм МВА (МНА) завершує свою роботу після скінченного числа кроків. Алгоритм МВА будує повну систему інваріантів і його часова складність пропорціональна величині $n \cdot m^{3,46}$, де n - $|A|$, m - $|R|$.

У першому параграфі п'ятого розділу розглядаються U-Y-програми, алгебрами даних яких є вільні групи. Основним інструментом при дослідженні таких програм є нільсенівські перетворення і нільсенівські системи твірних. Зокрема, задача про співвідношення розв'язується за допомогою алгоритму Нільсена, який був спроектований автором (разом з О.А. Летичевським і О.Б. Годлевським) і реалізований у системі алгебраїчного програмування АПС.

Більш складною виявилась задача про перетин. Із способу розв'язання задачі про співвідношення випливає, що задача про перетин у даному випадку зводиться до пошуку перетину скінченного числа нормальних дільників, які породжені скінченними множинами слів. Перша трудність, що лежить на шляху точного розв'язку задачі про перетин, полягає в тому, що ці нормальні дільники, взагалі кажучи, можуть не бути скінченнопородженими підгрупами групи $T_D(R)$, тобто не мати скінченних алгебраїчних базисів. У зв'язку з цим доводиться вводити додаткові умови, що накладають обмеження на підгрупи групи $T_D(R)$, - умови скінченної породженості нормальних підгруп, які складають множини співвідношень лінійних ланок програми. Такі обмеження, природно, приводять до більш бідних множин інваріантів, але дають практичний шлях для побудови таких множин або їх базисів.

Відомо, що перетин скінченного числа скінченнопороджених підгруп вільної групи сам є скінченнопородженою підгрупою вільної групи (теорема Хаусона). Але перше доведення цього факту не було конструктивним. Більш як через двадцять років А. В. Анісімовим було розглянуте питання про побудову відповідного алгоритму з позицій теоретичного програмування, завдяки чому було доведено існування алгоритму. Але цей алгоритм мав експоненціальну оцінку часової складності. Пізніше автором даної роботи був розроблений поліноміальний алгоритм побудови базису перетину двох скінченнопороджених вільних груп, характеристику якого дає таке твердження.

Теорема 5.1.2. Перетин скінченного числа скінченнопороджених вільних підгруп вільної групи є скінченнопородженою вільною підгрупою. Часова складність алгоритму побудови нільсенівської системи твірних для групи $H = G(P_1) \cap G(P_2)$ за множинами нільсенівських твірних P_1 і P_2 для підгруп пропорціональна величині $const \cdot N^5$, де $N = \max (L(P_1),\ L(P_2))$, а $L(P_i)$ - сумарна довжина множини твірних P_i, $i = 1,2$.

Таким чином, якщо множини ef(M,y) являють собою скінченнопороджені підгрупи вільної групи, то задача про перетин має конструктивний розв'язок. Але, як було зауважено вище, множини ef(M,y) не завжди є скінченнопородженими групами, і тому потрібно обмежуватись розглядом не всієї множини $C(P) = $ ef(M,y), а її скінченнопородженої підгрупи Gp(R). Властивість функції генерації ef у цьому випадку дає таке твердження.

Теорема 5.1.3. Функція ef є монотонною функцією.

Отже, в силу теорем 5.1.3 і 3.3.8 алгоритм МВА будує в цьому випадку максимально можливу множину інваріантів.

Для підвищення генеративної спроможності алгоритмів розглядаються такі способи поповнення базисів підгруп.

Нехай маємо базиси P_1 і P_2, позначимо через $P_1^{\sim}$ і $P_2^{\sim}$ множини, які одержані з цих базисів внаслідок поповнення.

а) Циклічне поповнення базисів P_1 і P_2 полягає в тому, що ці множини поповнюються циклічними перестановками слів, що їх складають.

б) Спряжене поповнення базисів P_1 і P_2 полягає в тому, що циклічно поповнені базиси $P_1^{\sim}$ і $P_2^{\sim}$ поповнюються словами, які є спряженими, тобто якщо в процесі застосувань нільсенівських перетворень одержане слово $xpx^{-1} \in H$, то $p \in H$, оскільки H - нормальний дільник. Спряжене поповнення базисів обчислюється за допомогою модифікованого алгоритму Нільсена. Ця модифікація полягає в тому, що порівняння довжин слів вихідного і одержаного в результаті застосування нільсенівського перетворення виконується тільки після того, як в одержаному слові вилучена спряженість (тобто після того, як це слово буде перетворено в циклічно нескорочуване). Алгоритм завершує свою роботу тоді, коли жодне перетворення не зменшує довжину слів і всі слова циклічно приведені.

в) Комутаторне поповнення базису P, одержаного в результаті роботи алгоритму перетину за спряжено поповненими базисами $P_1^{\sim}$ і $P_2^{\sim}$, полягає в тому, що множина P поповнюється словами, які є комутаторами вигляду $p^{-1}q^{-1}pq$, $p \in P_1$, $q \in P_2$, $i=1,\dots,m$, $j=1,\dots,k$. Виконуючи після цього побудову системи нільсенівських твірних для P, одержуємо базис для групи $H = H(P_1) \cap H(P_2)$. Слід зауважити, що коли є потреба, то поповнювати множину P можна як елементами вигляду $p^{-n}q^{-n}p^{n}q^{n}$, так і елементами вигляду $q^{-n}p^{-n}q^{n}p^{n}$, де n - довільне скінченне натуральне число.

Розглянуті апроксимації множин співвідношень є зростаючими, тобто такими, які підвищують генеративну силу описаних алгоритмів пошуку інваріантних співвідношень.

Підводить підсумок всього параграфа така теорема.

Теорема 5.1.4. Для будь-якої U-Y-програми, алгебра даних якої є вільною групою з умовою скінченної породженості для підгруп, алгоритм МВА завершує свою роботу через скінченне число кроків. Результат роботи алгоритму МВА не залежить від способу обходу станів U-Y-програми.

У другому параграфі п'ятого розділу розглядаються U-Y-програми, алгебри даних яких є скінченномірні лінійні афінні простори, а мова інваріантів розширюється до мови лінійних рівностей і лінійних нерівностей.

При розв'язанні задачі про співвідношення розглядаються два випадки. Припустимо, що базис $M' = M \cup \{u\}$ множини ef(M,u,y) скінченний. Представимо його у вигляді системи

$$Ar + B, \tag{5.2.3}$$

де $A=|\alpha_{ij}|$, а $r = (r_1,\dots,r_m)$ і $B = (\beta_1,\dots,\beta_m)$ - вектори-стовпці, $i = 1,2,\dots,n$, $j = 1,2,\dots,m$.

<u>*а) Невироджений випадок*</u> - випадок коли матриця T оператора присвоювання y має обернену матрицю, тобто $|T| \neq 0$.

Теорема 5.2.1. Якщо $|T| \neq 0$, то система $(AT^{-1})r + B = (AT^{-1})r < 0$ є породжуючою для $ef(M,u,y) = ef(M',:)$.

З цієї теореми випливає простий алгорим розв'язання задачі про співвідношення, характеристику якого дає

Наслідок 5.2.1. Часова складність алгоритму розв'язання задачі про співвідношення пропорціональна величині $O(N^3)$, де $N = \max\{m,n\}$.

б) Вироджений випадок - випадок, коли матриця T оператора присвоювання у - вироджена. Розглянемо матрицю

$$T' = \begin{vmatrix} \tau_{11} & \dots & \tau_{1m} & \gamma_1 \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \tau_{m1} & \dots & \tau_{mm} & \gamma_m \\ 0 \; 0 & \dots & 0 & 1 \end{vmatrix}.$$

Очевидно, що T' теж вироджена. Нехай $\lambda = (\lambda_1,\dots,\lambda_{m+1})$ - вектор-стовпчик, а T'' - матриця, транспонована до T'. Розглянемо систему рівнянь

$$T''\lambda = 0. \qquad (5.2.4)$$

Нехай $q < m$ - ранг T'', а

$$\begin{vmatrix} -1 & \dots & 0 & \lambda_{11} & \dots & \lambda_{1q} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & -1 & \lambda_{p1} & \dots & \lambda_{pq} \end{vmatrix} \qquad (5.2.4')$$

- матриця, що відповідає фундаментальній системі розв'язків (5.2.4), де $p = n - q + 1$. Тоді змінні $r_{i1},\dots,r_{ip}$ можна виразити через змінні $r_{ip+1},\dots, r_{im-1}, r_{im}, r_{im+1} = 1$:

$$\begin{cases} r_{i1} = \lambda_{11}r_{ip+1} + \dots + \lambda_{1q-1}r_{im} + \lambda_{1q} \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ r_{ip} = \lambda_{p1}r_{ip+1} + \dots + \lambda_{pq-1}r_{im} + \lambda_{pq}. \end{cases} \qquad (5.2.5)$$

Якщо в матриці T' i-ий рядок має вигляд $(0,\dots,0,1,0,\dots,0)$, тобто $\tau_{ij} = 0$ при $i \neq j$ і $\tau_{ii} = 1$, то відповідне λ_i в базис простору розв'язків системи (5.2.4) не включається, і тоді r_i буде однією із змінних $r_{ip+1},\dots,r_{im+1}$.

Нехай $I_1 = \{i_1,\dots,i_p\}$, $I_2 = \{i_{p+1},\dots,i_m,i_{m+1}\}$. Визначимо матрицю

$$T' = |\tau_{ij}'| \qquad (5.2.6)$$

таким чином:

$$\tau_{ij}' = \begin{cases} \lambda_{kl-p}, \text{ якщо } i = i_k \in I_1,\ j = i_l \in I_2; \\ 1, \text{ якщо } i = j \in I_2; \\ 0, \text{ інакше}, \end{cases}$$

де λ_{kl-p} - коефіцієнти системи (5.2.5).

Розв'язок задачі про співвідношення, коли матриця T' має вигляд (5.2.6), випливає з такої теореми.

Теорема 5.2.2. Якщо матриця T' оператора у має вигляд (5.2.6), а система (5.2.3) є породжуючою для $M' = M \cup \{u\}$, то $ei(M',y)$ породжується системою, яка одержана об'єднанням фундаментальної $(E_{i1}+\ldots+E_{ip})$ - згортки системи (5.2.3) з системою (5.2.5).

Розв'язання задачі про перетин зводиться до такої задачі. Нехай M_1 і M_2 - деякі множини лінійних нерівностей, що породжені відповідно системами

$$A_1\Gamma + B_1 < 0 \quad і \quad A_2\Gamma + B_2 < 0,$$

де $A_1 = |\alpha_{ij}|$, $i = 1,\ldots,m$, $A_2 = |\alpha_{ij}|$, $i = n+1,\ldots,n+k$, $B_1 = (\beta_1,\ldots,\beta_n)$, $B_2 = (\beta_{n+1},\ldots,\beta_{n+k})$.

Розглянемо матриці

$$\bar{A}_1 = \begin{vmatrix} \alpha_{11} & \ldots & \alpha_{1m} \\ \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots \\ \alpha_{n1} & \ldots & \alpha_{nm} \\ 0 & \ldots 0 & -1 \end{vmatrix}, \qquad \bar{A}_2 = \begin{vmatrix} \alpha_{n+1} & \ldots & \alpha_{n+1m} \\ \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots \\ \alpha_{n+k1} & \ldots & \alpha_{n+km} \\ 0 & \ldots 0 & -1 \end{vmatrix},$$

вектори $\bar{B}_1 = (\beta_1,\ldots,\beta_n,-1)$, $\bar{B}_2 = (\beta_{n+1},\ldots, \beta_{n+k},-1)$, вектори-стовпчики $\Gamma_1 = (\gamma_1,\ldots,\gamma_n,\gamma_0)$, $\Gamma_2 = (\gamma_{n+1},\ldots\gamma_{n+k},\gamma_{n+k+1})$ і 0_i - нульовий вектор-стовпчик довжиною i.

Складемо систему

$$\begin{cases} \bar{A}_1\cdot\Gamma_1 - \bar{A}_2\cdot\Gamma_2 = 0 \\ \bar{B}_1\cdot\Gamma_1 - \bar{B}_2\cdot\Gamma_2 = 0 \\ \Gamma_1 > 0 \\ \Gamma_2 > 0, \end{cases} \tag{5.2.7}$$

і нехай Φ - фундаментальна система розв'язків (5.2.7). Позначимо через P множину всіх нерівностей, які є лінійними комбінаціями нерівностей системи $A_1\Gamma + B_1 < 0$ або системи $A_2\Gamma + B_2 < 0$ з коефіцієнтами $(\gamma_1,\ldots,\gamma_n,\gamma_0,\ldots,\gamma_{n+k+1}) \in \Phi$.

Теорема 5.2.3. Множина P є породжуючою множиною для $M_1 \cap M_2$.

З цієї теореми випливає алгоритм розв'язання задачі про перетин. Але в силу того, що побудова фундаментального набору розв'язків системи лінійних нерівностей має експоненціальну оцінку часової складності, часова складність всього алгоритму розв'язання задачі про перетин теж експоненціальна за сумарним числом нерівностей в системах P_1 і P_2.

Розв'язання задачі про стабілізацію у випадку мови лінійних рівностей і нерівностей зустрічається з деякими складностями, які пов'язані з тим, що послідовність $N_a^{(0)} \supset N_a^{(1)} \supset \ldots \supset N_a^{(n)} \supset \ldots$ співвідношень у заданому стані а може бути нескінченною. Для забезпечення скінченності цієї послідовності використовується операція звуження ∇, за допомогою якої обчислюється

$$N_a^{(i+1)} = N_a^{(i)} \nabla \left(\bigcap_{j=1}^{k} ef(N_a^{(i)}, u_l, y_l) \right).$$

Операція ∇ визначається так: нехай P^* - поліедр розв'язків множини співвідношень P, тоді $M \nabla P$ - це опукла оболонка множини всіх тих елементів із базису M, які виконуються в будь-якій точці з базису поліедра розв'язків P^*. Ця операція забезпечує стабілізацію, тобто окінченність, послідовності

$$N_a^{(0)} \supseteq N_a^{(1)} \supseteq \ldots \supseteq N_a^{(n)} \supseteq \ldots$$

Неважко переконатися в тому, що із включення $N \subseteq N'$ випливає включення $ef(N,u,y) \subseteq ef(N',u,y)$. Отже, якщо в алгоритмі МНА використати за початкову множину умов більш багаті множини співвідношень, ніж тотожності алгебри $T_D(R)$, то в результаті роботи МНА повинні побудуватися більш багаті множини інваріантів N_a, $a \in A$. Візьмемо за початкові множини $N_a^{(0)}$ множини інваріантів, які дає алгоритм МВА, і застосуємо алгоритм МНА. В результаті одержимо послідовність множин інваріантів

$$N_a^{(0)} \subseteq N_a^{(1)} \subseteq N_a^{(2)} \subseteq \ldots .$$

Гранична множина $N_a = \bigcup_{i=0}^{\infty} N_a^{(i)}$ дає добре наближення для множини всіх інваріантів стану a. Послідовність $N_a^{(0)} \subseteq N_a^{(1)} \subseteq \ldots \subseteq N_a^{(n)} \subseteq \ldots$ може бути неокінченною, але, обмежившись певною точністю, можна зупинитися на будь-якому кроці.

Слід зауважити, що в даному випадку генеруючу спроможність алгоритму МВА можна посилити, якщо операцію звуження при обчисленні $N_a^{(i)} = N_a^{(i-1)} \nabla N$ починати застосовувати з деякого n-го кроку ($n > 3$), а не з другого, як в алгоритмі МВА.

Питання повноти множин інваріантів, що будуються таким чином, випливають з таких тверджень.

Лема 5.2.2. Нехай P_1 і P_2 - довільні опуклі множини в просторі D^m, $P_1 \sqcup P_2$ - їх опукла оболонка, y - лінійне афінне перетворення D. Тоді $y(P_1 \sqcup P_2) = y(P_1) \sqcup y(P_2)$.

Нехай C_1 і C_2 (C_1^* і C_2^*) - конуси (поліедри розв'язків) систем лінійних нерівностей M_1 і M_2 відповідно, а $C_1 \sqcup C_2$ - їх опукла оболонка. Очевидно, що $C_1 \sqcup C_2$ є лінійним опуклим конусом, множина твірних якого є об'єднання множин твірних конусів C_1 і C_2. Нехай P' означає конус системи лінійних нерівностей, поліедр розв'язків якої дорівнює P.

Лема 5.2.3. Справедливі рівності

а) $(C_1 \cap C_2)^* = C_1^* \sqcup C_2^*$; б) $(P_1 \sqcup P_2)' = P_1' \sqcup P_2'$.

Теорема 5.2.4. $ef(M_1 \cap M_2, y) = ef(M_1, y) \cap ef(M_2, y)$.

Наслідок 5.2.2. $ef(M_1 \cap M_2, u, y) = ef(M_1, u, y) \cap ef(M_2, u, y)$.

Теорема 5.2.4 і наслідок 5.2.2 показують, що при роботі алгоритмів МВА і МНА співвідношення втрачаються тільки за рахунок операції звуження.

У шостому розділі описані деякі застосування теорії програмних інваріантів до оптимізації програм і проектування ефективних алгоритмів.

Перший параграф присвячений редукції U-Y-програм на основі інваріантних співвідношень з метою оптимізації. Основним джерелом таких перетворень програми виступають

а) тотожні та квазітотожні співвідношення алгебри алгоритмів;

б) інваріантні співвідношення програми.

Якщо співвідношення а) перетворюють регулярний вираз, за допомогою якого задається програма, то співвідношення б) перетворюють складові самого регулярного виразу (виконують редукцію виразу).

Редукція U-Y-програм на основі інваріантних співвідношень зводиться до таких перетворень. Оператор y (умова u) називається *приведеним відносно множини інваріантів* N, якщо праві частини оператора y (терми умови u) приведені відносно N, тобто всі вирази оператора y (умови u) мають найменшу вагу операцій відносно множини N.

Визначаються такі перетворення на U-Y-програмі:

T_1: вилучення умов, які константні відносно множини інваріантів;

T_2: приведення операторів і умов відносно множин інваріантів;

T_3: вилучення вироджених операторів присвоювання $(r_i := r_i)$.

Всі ці перетворення зберігають функціональну еквівалентність вихідної і одержаної U-Y-програм A і A' відносно початкової множини інваріантів N_0. *U-Y-програма* A називається *приведеною* відносно сімейства інваріантів $\{N_a^{Gen}\}$ тоді і тільки тоді, коли в A вилучені всі константні умови і вироджені оператори присвоювання, а решта операторів і умов приведені відносно $\{N_v^{Gen}\}$, тобто жодне з перетворень T_1 – T_3 не застосовне до A. U-Y-програма A називається *повністю приведеною відносно генератора інваріантів Gen*, якщо вона приведена відносно сімейства

$$\{N_v^{Gen} = \bigcap_{s \in path(v)} ef(N, y_s)\}.$$

Нехай $\{A(N_0)\}_{Gen}$ означає клас U-Y-програм, у яких початкова множина умов є N_0, а будь-які дві U-Y-програми з цього класу відрізняються тим, що в одній з них чи в обох є константні умови або неприведені вирази, або вироджені оператори присвоювання.

Процес приведення U-Y-програми A з початковою множиною N_0 відносно $\{N_v^{Gen}\}$ будемо називати *редукцією* U-Y-програми A, а одержана в результаті цього U-Y-програма буде називатися залишковою. Залишкова U-Y-програма позначається $Red_{Gen}(A, N_0)$ або просто Red(A), коли відомо відносно якої множини інваріантів виконується редукція. Має місце таке твердження.

Теорема 6.1.1. Для будь-якої U-Y-програми A із $\{A(N_0)\}_{Gen}$, U-Y-програма Red(A) збігається з повністю приведеною U-Y-програмою для A тоді і тільки тоді, коли сімейство $\{N_v^{Gen}\}$ є повним.

Як випливає з результатів розділів 3 і 4, є такі генератори інваріантів, які дають повні множини інваріантів. Отже, справедлива така теорема.

Теорема 6.1.2. Якщо N_0 - нормальна конгруенція і алгебра даних U-Y-програми з абсолютно вільною Ω-алгеброю або комутативним Ω-групоїдом (комутативним групоїдом) із скороченнями і нулем відносно комутативної операції або Ω-напівгрупою (напівгрупою) з початковою множиною співвідношень N_0, що задовольняє умові фінітарності, або вільною комутативною напівгрупою, або абелевою групою, або афінним лінійним простором і генератор інваріантів Gen реалізується за допомогою методу верхньої апроксимації для мови рівностей, то множини інваріантів $\{N_v^{Gen}\}$ U-Y-програми A є повними і залишкова програма Red(A) є повністю приведеною.

Отже, з теорем 6.1.1 і 6.1.2 випливає, що при виконанні їх умов для U-Y-програми A над кожною із перелічених алгебр залишкова U-Y-програма Red(A) може бути побудована в деякому розумінні найкращим чином.

У другому параграфі пропонується методика оптимізації програм з процедурами, яка виникла в рамках методу формалізованих технічних завдань, де основою перетворень програм служить техніка застосування співвідношень. Методика базується на використанні інваріантних співвідношень, які стосовно даного випадку виступають в ролі задання контексту виклику тієї чи іншої процедури в основній програмі. Необхідно зауважити, що в розглянутих випадках інваріантні співвідношення, які задають контекст виклику процедури, виступають в ролі обмежень застосувань процедури, що викликається, і їх можна розглядати як рівняння, що задають звуження області застосування цієї процедури.

Основні оптимізаційні перетворення процедур зводяться до таких.

а) Приведення процедур з врахуванням входу. Нехай N - L - деяка множина інваріантів на вході процедури P і O_P - тіло процедури P. *N-приведенням* P_N процедури P називається залишкова U-Y-програма Red(P,N), отримана в результаті редукції O_P відносно $\{N_v^{Gen}\}$. В багатьох випадках, особливо коли O_P має вигляд умовного оператора і умова в цьому операторі константна відносно відповідної множини інваріантів, N-приведення процедури P суттєво простіше вихідної процедури.

б) Введення додаткових точок входу в процедури. Введення в U-Y-програму додаткових процедур веде до збільшення об'ємів пам'яті, яку займає програма. Але в окремому випадку за рахунок введення додаткових точок входів у тіло процедури можна одержувати нові процедури, які є проміжними за часом виконання між вихідними процедурами і їх N-приведеннями. Введення додаткових точок входів у тіло процедури з точки зору затрат пам'яті значно дешевше, ніж введення нових процедур.

Необхідно зауважити, що така техніка пошуку в тілі процедури точок, куди в процесі оптимізації слід вставити точку входу пов'язана з досить складним аналізом. Тому далі пропонується більш проста, хоча і менш універсальна техніка оптимізації, основана на використанні інваріантних співвідношень.

в) Процедури із збереженням входів. Нехай R_1 - множина вхідних змінних процедури P (сюди відносяться також глобальні по відношенню до неї змінні, що використовуються в тілі P), а R_2 - множина вихідних змінних. Процедура P називається процедурою із збереженням входів, якщо за будь-яких вихідних значень вхідних змінних вони не змінюються в результаті звернення до цієї процедури. Повторне звернення до процедури із збереженням входів, очевидно, еквівалентне однократному зверненню. Отже, якщо в програмі зустрічається послідовність операторів call P; Q_1;...; Q_n; call P, де всі Q_i (i-1,2,...,n) не змінюють вхідних і вихідних значень такої процедури P, то другий оператор виклику процедури P надлишковий.

Виграш від такого оптимізаційного перетворення визначається часом виконання процедури P і, взагалі кажучи, може бути значним, особливо коли оператор call P знаходиться на ділянці програми, що багаторазово повторюється.

г) Приведення процедур з врахуванням виходу. Описані вище прийоми оптимізації базувались на використанні (у вигляді інваріантних співвідношень) інформації, що мала місце в точках входу в оператори звернення до процедур. Ситуація, яка розглядається, в деякому плані протилежна тій, що була описана вище. Основою для оптимізаційних перетворень процедури P служить інформація про використання змінних, значення яких виробляються при зверненні до цієї процедури. Якщо не всі змінні, що вироблені процедурою за даного звернення до неї, використовуються в тому, чи іншому контексті, то процедуру P можна спростити шляхом вилучення з її тіла тих операторів, які вироблять значення таких змінних.

У третьому параграфі шостого розділу розглядається процес трансформаційного синтезу ефективних алгоритмів для мінімізації скінченного ациклічного автомата без виходів (лінійний алгоритм від числа переходів в автоматі) і уніфікації як скінченних, так і нескінченних термів в абсолютно вільній алгебрі термів (майже лінійний алгоритм, тобто часова складність виражається величиною n·G(n), де n - довжина термів, а G(n) - функція обернена до функції Аккермана). Перший з цих алгоритмів, як випливає з розділу 4, застосовується для вирішення проблеми пошуку спільних підвиразів у заданому виразі, а отже, і розв'язку задачі про співвідношення.

<u>У сьомому, останньому розділі</u> описується реалізація в системі алгебраїчного програмування АПС деяких алгоритмів комбінаторної теорії груп. Основними з таких алгоритмів є алгоритм побудови канонічної форми елемента у вільній групі і алгоритм Нільсена - алгоритм побудови базиса скінченнопородженої підгрупи вільної групи. На базі реалізації цих алгоритмів побудовані і реалізовані алгоритми розв'язання таких проблем:

- включення слова в задану скінченнопороджену підгрупу вільної групи;
- включення для скінченнопороджених підгруп вільної групи;
- ізоморфізму, гомоморфізму, автоморфізму та тотожності для скінченнопороджених підгруп вільної групи.

Реалізовані також деякі прості алгоритми представлення скінченнопороджених напівгруп і скінченних алгебр скінченними автоматами.

ОСНОВНІ РЕЗУЛЬТАТИ РОБОТИ

В дисертаційній роботі на базі методів сучасної загальної алгебри та теорії автоматів розроблена загальна теорія статичного аналізу програм, і на цій основі досліджена і побудована ціла низка алгоритмів пошуку множини інваріантних співвідношень для програм над різноманітними класичними алгебрами даних. Характерною відмінністю від існуючих методів статичного аналізу є те, що побудова програмних інваріантів ведеться з урахуванням властивостей алгебри даних, над якою працює програма.

Основними науковими результатами дисертації є такі:

1. Запропоновано та розроблено алгеброавтоматний підхід до розв'язання проблеми побудови множини інваріантних співвідношень для програм з урахуванням властивостей алгебри даних. Розроблено два основних методи пошуку програмних інваріантів - метод верхньої і метод нижньої апроксимації, а також побудовані відповідні алгоритми для цих методів.

2. Досліджено властивості методів верхньої і нижньої апроксимації, а також відповідних їм алгоритмів для випадку мови рівностей. Сформульовані основні задачі теорії програмних інваріантів для цієї мови і встановлені умови, при виконанні яких алгоритм верхньої апроксимації дає повні множини програмних інваріантів. Доведено також, що у випадку порушення умов повноти результат роботи алгоритму верхньої апроксимації є максимальною нерухомою точкою розв'язку відповідної системи рівнянь.

3. Встановлено, що алгоритм верхньої апроксимації дає повні множини програмних інваріантів для програм, алгебра даних якої є однією з таких алгебр:

- абсолютно вільною Ω-алгеброю;
- комутативним Ω-групоїдом (комутативним групоїдом) з нулем і скороченнями відносно комутативної операції;
- Ω-напівгрупою (напівгрупою) з умовою фінітарності;
- скінченномірним лінійним афінним простором;
- вільною комутативною напівгрупою;
- вільною абелевою групою.

4. Показано, що в загальному випадку алгоритм верхньої апроксимації дає хоча і неповну множину інваріантних співвідношень для програм, алгебра даних якої є вільною групою, але ця множина

являє собою максимальну фіксовану точку розв'язку відповідної системи рівнянь.

Встановлено, що алгоритм верхньої апроксимації для випадку мови лінійних рівностей і лінійних нерівностей у програмах, алгебра даних яких є скінченномірним афінним простором, за відповідних обмежень дає повну множину інваріантних співвідношень.

5. На базі теорії програмних інваріантів запропоновані методика оптимізаційних перетворень програм з процедурами, а також методика оптимізаційних перетворень при трансформаційному методі проектування ефективних алгоритмів.

6. Розроблено серію ефективних алгоритмів обчислення конгруентних замикань виразів в скінченновизначених алгебрах, таких, як

- Ω-алгебра термів;
- комутативний Ω-групоїд (комутативний групоїд);
- Ω-напівгрупа (напівгрупа) з умовою фінітарності;
- комутативна Ω-напівгрупа;
- абелева Ω-группа.

7. Одержані такі результати загальноматематичного характеру:

а) спадна (зростаюча) послідовність нормальних конгруенцій скінченнопородженої Ω-алгебри задовольняє умові обриву ланцюгів, а отже, і умовам максимальності і індуктивності;

б) перетин скінченного числа скінченнопороджених конгруенцій, які зберігають ациклічність термів, у випадку

- абсолютно вільної Ω-алгебри;
- комутативного Ω-групоїда;
- комутативного групоїда;
- Ω-напівгрупи з умовою фінітарності

є скінченнопородженою конгруенцією.

Основні результати дисертації опубліковані у таких роботах:

1. Кривой С.Л., Ракша С.Г. Поиск инвариантных линейных зависимостей в программах // Кибернетика.-1984.- N 6.- С. 23-28.
2. Кривой С.Л. Об эффективности некоторых алгоритмов комбинаторной теории групп // Кибернетика.-1985.- N 3.- С. 9-14.
3. Системное и математическое обеспечение многопроцессорного вычислительного комплекса ЕС 1766 / Под ред. Ю.В. Капитоновой. -М.: ВВИА им. Н.Е. Жуковского,1986.-390 с.
4. Годлевский А.Б., Кривой С.Л. Трансформационный синтез некоторых алгоритмов с учетом дополнительных спецификаций / Кибернетика.- 1986.- N 6.- С. 37-43.

5. Годлевский А.Б., Кривой С.Л. О проектировании эффективных алгоритмов приведения автоматов для некоторых отношений эквивалентности // Кибернетика.- 1989.- N 6.- С. 54-61.
6. Итеративные методы анализа программ / Годлевский А.Б., Капитонова Ю.В., Кривой С.Л., Летичевский А.А. // Кибернетика. -1989. -N 2. -С. 10-19.
7. Итеративные методы анализа программ. Равенства и неравенства / Годлевский А.Б., Капитонова Ю.В., Кривой С.Л., Летичевский А.А. // Кибернетика.- 1990.- N 3.- С. 1-10.
8. Кривой С.Л. Алгоритмы минимизации ациклических автоматов и идентификация образцов в термах // Кибернетика.- 1991.- N 3.- С. 11-16.
9. Годлевский А.Б., Кривой С.Л. Редукция программ с учетом дополнительных спецификаций // Кибернетика и системный анализ.-1991.- N 5. - С. 57-67.
10. Кривой С.Л. Алгоритмы вычисления конгруэнтных замыканий конечных автоматов и некоторые их приложения // Кибернетика и системный анализ.- 1994.- N 1.- С. 34-44.
11. Летичевский А.А., Кривой С.Л. О реализации алгоритма Нильсена в системе алгебраического программирования АПС-1 // Кибернетика.- 1994.- N 5.- С. 126-133.
12. Кривой С.Л. О реализации некоторых алгоритмов комбинаторной теории групп в системе АПС-1 // Кибернетика.- 1994.- N 6.- С. 166-175.
13. Луцький Г.М., Кривий С.Л., Печурін М.К. Основи дискретної математики (учбовий посібник для студентів технічних вузів). -Київ: ВІПОЛ, 1995.- 252 с.
14. Кривой С.Л. О базисе пересечения множеств равенств термов в свободной алгебре с соотношением коммутативности // Доп. НАН України.- Київ.- 1996.- N 4.- С. 61-64.
15. Кривой С.Л. Обобщение алгоритмов поиска инвариантных соотношений в программах над алгеброй термов // Кибернетика и системный анализ. -1996.-N 1.- С.24-36.
16. Кривой С.Л. О вычислении конгруэнтных замыканий на термах // Кибернетика и ситемный анализ.- 1997.- N 4.- С.121-132.
17. Кривой С.Л. Проблема унификации в эквациональных теориях // Кибернетика и системный анализ.-1997.- N 6.- С. 73-102.

Особистий внесок. Всі результати, наведені в дисертаційній роботі, належать повністю дисертанту і одержані ним самостійно. Зокрема, в роботі [1] авторові належить розробка основних алгоритмів та їх обгрунтування, в монографії [3] - сторінки 193-203. В роботах [6,7] особистий вклад автора полягає в огляді власних результатів, які в повному обсязі викладені в розділах 3, 4 і 5. В роботах [4,5] авторові належать розробка алгоритму мінімізації для скінченних ациклічних автоматів і основна ідея та алгоритм приведення скінченних автоматів для автоматної еквівалентності, а також встановлення властивостей функції злиття для x-еквівалентності. В роботі [11] - описана в розділі 7 реалізація алгоритму Нільсена і в роботі [13] авторові належать розділи 1, 5, 6.

КРИВОЙ С.Л. Итеративные методы анализа процедурных программ. Диссертация на соискание ученой степени доктора физико-математических наук по специальности 01.05.01 - теоретические основы информатики и кибернетики. Институт кибернетики им. В.М. Глушкова НАН Украины, Киев, 1997.

Защищается рукопись на основе 41 работы. Разработана общая математическая теория анализа процедурных программ с одноуровневой памятью, ориентированная на генерацию из текста программы ее инвариантов с учетом свойств алгебры данных, над которой работает рассматриваемая программа.

Krivoi S.L. Iterative methods for analyse of procedural programs. Doctoral thesis of physics and mathematics. 01.05.01 - theoretical foundations of informatic and computer science. Glushkov Institute of Cybernetics of the National Academy of Sciences of Ukraine. Kiev, 1997.

The thesis defends a manuscript based on 41 scientific works. A general mathematical theory of data flow analysis for procedural programs with one level of memory and the properties of data algebras is developed.

Ключові слова. Вільні алгебри, U-Y-програми, конгруенції, гомоморфізми, ізоморфізми, інваріантні співвідношення, редукція, оптимізаційні перетворення програм.

Підп. до друку 24.10.97. Формат 60×84/16. Папір офс. Офс. друк. Ум. друк. арк. 1,86. Ум. фарбо-відб. 2,09. Обл.-вид. арк. 2,0. Тираж 100 прим. Зам. 416.

Редакційно-видавничий відділ з поліграфічною дільницею
Інституту кібернетики імені В. М. Глушкова НАН України
252022 Київ 22, проспект Академіка Глушкова, 40

Ab 38.811