Академія наук України
Інститут кібернетики імені В.М. Глушкова

На правах рукопису

БАКУЛІН Олександр Володимирович

УДК 681.3.06

РОЗВИТОК МЕТОДУ ТА ІНСТРУМЕНТАРІЮ БАГАТОРІВНЕВОГО ДОКАЗОВОГО ПРОЕКТУВАННЯ ПРОГРАМ

05.13.11 – математичне та програмне забезпечення обчислювальних машин, комплексів, систем та мереж

А В Т О Р Е Ф Е Р А Т
дисертації на здобуття вченого ступеня кандидата фізико-математичних наук

Київ 1992

Роботу виконано в Інституті
імені В.М.

Науковий керівник: доктор технічних наук
ЦЕЙТЛІН Г.Є.

Офіційні опоненти: член-кореспондент АН України,
доктор фізико-математичних наук
Летічевський О.А.

кандидат фізико-математичних наук
Каюров В.Ю.

Провідна установа: Київський державний університет
ім. Т.Г. Шевченка

Захист відбудеться "12" лютого 1993 р. о 14 годині на засіданні спеціалізованої ради Д 016.45.01 при Інституті кібернетики імені В.М. Глушкова АН України за адресою:
252207 Київ 207, проспект Академіка Глушкова, 40

З дисертацією можна ознайомитись в науково-технічному архіві Інституту

Автореферат розіслано "18" грудня 1992 року

Учений секретар
спеціалізованої ради
кандидат фізико-математичних наук В.Ф.Синявський

ЗАГАЛЬНА ХАРАКТЕРИСТИКА РОБОТИ

Актуальність проблеми. Створення надійного програмного забезпечення (ПЗ) залишається однією з нерозв'язаних проблем сучасного програмування. Істотний вплив на надійність мають помилки проектування, усунення яких, особливо на заключних етапах розробки, призводить до значних витрат. На протязі багатьох років розвиваються і вдосконалюються підходи, при яких програмування розглядається як певний вид математичної діяльності: конструктивне визначання функцій в функціональному і композиційному програмуванні, побудова аксіоматичних систем в алгебраїчному, концептуальному, логічному програмуванні, розробка математичних моделей предметних областей в методах формалізованих технічних завдань, VDM, Z, формальні перетворення математичних описань в трансформаційному програмуванні. В рамках формалізованих методів програмування доцільно розглядати і задачу перевірки проектів програм.

Один з напрямів сучасного програмування, заснований на застосуванні алгебраїчних методів, сформувався як результат розвитку теорії систем алгоритмічних алгебр (САА) В.М. Глушкова та її застосувань. Апарат САА при відповідній інтерпретації носіїв та операцій використовується для багаторівневого проектування програм; модифіковані системи алгоритмічних алгебр (САА/М) додатково поставляють засоби описання паралелізму; в алгебрі структур даних (АСД) - переінтерпретації САА/М - визначений гнучкий та виразний формалізм для описання і багаторівневого проектування структур даних. Особливість розглядуваного математичного апарату полягає в можливості компактного структурованого запису схем програм і даних у вигляді формул, для яких відпрацьовано техніку тотожніх перетворень.

Апарат САА/М покладено в основу методу багаторівневого

структурного проектування програм (БСПП), який поєднує алгебраїчні методи з формальними моделями мов. Інструментарій БСПП - система МУЛЬТИПРОЦЕСИСТ - застосовувався при розв'язанні задач різних класів і продемонстрував підвищення надійності та скорочення часу відладки розроблюваних програм. Вбачається доцільним розвиток методу БСПП в орієнтації на доказове проектування, розповсюдження його на початкові етапи життєвого циклу ПЗ. Звідси необхідність створення відповідного математичного апарату, який можна отримати внаслідок інтегрування САА/М та АСД.

Мета дисертаційної роботи полягає в побудові теоретичних та інструментальних засобів, орієнтованих на доказове багаторівневе проектування, що передбачає створення на основі інтегрування САА/М та АСД математичної моделі для подання проектів програм, формалізацію поняття правильності проекту і побудову математичного апарату для її формального обгрунтування, реалізацію програмних засобів для підтримки доказового проектування.

Наукова новизна роботи полягає в наступному:

- побудовано логіко-алгебраїчний апарат, орієнтований на багаторівневе доказове проектування програм, на основі якого:

1) розвинуто підхід до розробки проектів програм та визначення семантики абстрактних перетворювачів даних, які застосовуються при проектуванні;

2) формалізовано поняття проекту програми та його правильності і розроблено методи аналізу проектів програм: тестування, верифікації, трасування (символічного виконання);

3) побудовано числення, призначене для формального доведення правильності проектів програм; доведено теореми про повноту та несуперечність числення, розроблено стратегії верифікації проектів програм;

- запропоновано заснований на використанні створеного матема-

тичного апарату метод багаторівневого доказового структурного проектування програм (БСПП/Д), що поєднує багаторівневу модель життєвого циклу ПЗ із обгрунтуванням проектів на кожному з рівнів у відповідності з запропонованими методами тестування, верифікації, трасування;
- в рамках створення програмного забезпечення методу БСПП/Д сформовано понятійну основу мови формалізованих специфікацій та проектування (САА-специфікацій) і розроблено проект інструментарію, який оформлено засобами цієї мови.

Практична цінність. Метод БСПП/Д, запропонований в дисертації, може використовуватись при розробці реальних програмних систем, що сприяє підвищенню надійності створюваного ПЗ. Це забезпечується використанням запропонованих методів обгрунтування правильності проектів програм та мови САА-специфікацій, наділеної розвинутими механізмами структуризації та декомпозиції, практична корисність якої була продемонстрована при побудові інструментарію БСПП/Д.

Простота та наглядність використовуваних засобів в поєднанні з достатньою математичною строгістю основних концепцій створюють передумови для використання методу БСПП/Д при вивченні формалізованих методів програмування.

Інструментарій методу БСПП/Д реалізовано в рамках розвитку РИТМ-технології, що проводиться у відповідності з Білоруською республіканською комплексною науково-технічною програмою "Інформатика" (номер держ. реєстрації 01.90.0 036397).

Публікації та апробація роботи. Приведені в роботі результати отримані автором самостійно і доповідались на Всесоюзних семінарах "Параллельное программирование и высокопроизводительные системы" (Бердянськ, 1986 р., Алушта, 1988 р., Планерське, 1989 р.) II-ій Всесоюзній конференції по прикладній логіці (Новосибірськ, 1988 р.), на семінарі відділу автоматизації програмування в Інституті кібернетики ім. В.М.Глушкова

АН України (1992 р.).

По темі дисертації опубліковано 9 робіт.

Структура і об'єм роботи. Робота складається із вступу, чотирьох розділів, висновків, списку літератури із 127 найменувань та двох додатків. Використано 2 малюнки та 3 таблиці. Основний машинописний текст (без додатків) складає 115 сторінок.

КОРОТКИЙ ЗМІСТ РОБОТИ

У вступі обгрунтовується актуальність проблеми, визначається мета дисертації. Дається означення та коротка характеристика формальних методів (ФМ) програмування. Приведено основні результати дисертації.

В першому розділі розглядаються мови формальних специфікацій (МФС), формулюються вимоги, яким вони повинні задовільняти. Проводиться огляд засобів САА/М та АСД з метою наступного розвитку на їх основі формалізованого методу програмування. Наведено означення основних понять САА/М та АСД, аналізуються можливості, пов'язані з їх використанням при розробці ПЗ. Обгрунтовується спосіб інтегрування САА/М та АСД, обумовлений їх застосуванням при доказовому програмуванні.

В дисертації визначено наступні вимоги до МФС: 1) наявність засобів структуризації та декомпозиції; 2) підтримка різних методів (модельних, аксіоматичних) специфікації та можливість вибору найбільш придатного в кожному конкретному випадку; 3) наявність асоційованої з МФС формальної системи, необхідної для дослідження властивостей специфікації; 4) використання в специфікаціях неформальних пояснень.

Модифікована система алгоритмічних алгебр (САА/М) - це трійка $\langle U,V;\Omega\rangle$, де елементи множини U (оператори) - відображення інформаційної множини M (сукупності станів аб-

страктної обчислювальної машини) в себе, елементи множини V (умови) - відображення M в тризначну множину логічних значень Bool, а сигнатура операцій Ω включає операції: композиція - A*B - послідовне виконання операторів A і B; α-диз'юнкція - (α:A\B) - конструкція типу if α then A else B; α-ітерація - {α:A} - конструкція типу while ¬α do A; фільтр - [α] - конструкція типу if α then skip else abort та інш. (крім goto). Алгоритм сортування вставками, наприклад, в CAA/M записується так, як показано в рядку 1 таблиці 1 (стор. 10), де INIT - оператор, що встановлює вказівник ! на початок масиву, SHF - оператор зсуву вказівника ! вправо по масиву, INS - оператор встановлення елемента, розміщеного зліва від вказівника !, у вже впорядковану частину масиву таким чином, щоб впорядкованість зберігалась, Endarr - умова, істинна, коли ввесь масив оброблено.

До основних понять алгебри структур даних (АСД) належать: конфігурація - ланцюжок символів деякого алфавіту, об'єкт - множина конфігурацій, застосування об'єкта (в режимі розпізнавання) - перевірка належності конфігурації об'єкту. Носіями в АСД вважаються об'єкти і умови, визначені на (розмічених спеціальними символами) конфігураціях. Сигнатура операцій АСД і CAA/M співпадає, але у випадку АСД операції визначено на об'єктах. Щоб задати в АСД структуру ланцюжка, необхідно вказати формулу, яка є представленням об'єкта, що розпізнає цей ланцюжок. Наприклад, формула із таблиці 1 (рядок 3), де конфігурації об'єкта Real - дійсні числа, задає числову послідовність (масив). За означенням, на кожному кроці α-ітерації розпізнається одна конфігурація об'єкта Real, і процес завершується, коли буде розпізнано послідовність конфігурацій із Real (стане істинною умова eofarr). Формули АСД подібним чином дозволяють визначати досить складні структури даних.

В орієнтації на доказове проектування пропонується переінтерпретувати інформаційну множину САА/М, використовуючи для цього засоби АСД. Це забезпечить: структурованість, строгість описання та можливість верифікації проектів програм; можливість декомпозиції розв'язуваних задач внаслідок застосування ієрархічно зв'язаних формул САА/М і АСД; легкість навчання, обумовлену подібністю засобів подання алгоритмів і даних.

В другому розділі в результаті інтегрування САА/М і АСД, розвинення їх зображальних можливостей на основі розширення нотації, уточнення існуючих та введення нових понять будується логіко-алгебраїчна модель, орієнтована на створення багаторівневих проектів програм. Запропонований математичний апарат поставляє засоби для описання і багаторівневого проектування алгоритмів і даних, а також спеціальні логічні засоби для описання семантики абстрактних примітивів, які використовуються при проектуванні. Розробляється концепція к-інтерпретації як інструмента означення функціональних специфікацій, формалізується поняття проекту програми і встановлюється порядок розробки проектів. Формується понятійна основа мови специфікацій і проектування програм.

Інструментом багаторівневого проектування алгоритмів і даних є сукупності послідовно уточнюваних формул САА/М і АСД. Вони утворюють операторні та об'єктні схеми, які в дисертації означаються таким чином. Об'єктною схемою називається сукупність рівностей виду $O_i=F_i(i=1..n)$, де O_i - об'єкти, F_i - формули АСД, такі що в лівій частині кожної рівності (крім першої) може зустрічатись лише такий об'єкт O_i, який вже з'являвся в формулі (формулах) попередніх рівностей. Тобто, кожна наступна рівність конкретизує за допомогою формули АСД об'єкт із попередньої рівності, що забезпечує багаторівневість подання. Аналогічно дається означення опера-

торної схеми, але з додатковою вимогою відсутності рекурсії. При цьому оператори розглядаються як відображення на множині кортежей помічених конфігурацій АСД, які називаються к-конфігураціями. Проведене таким чином інтегрування САА/М і АСД забезпечує можливість взаємопов'язаного описання алгоритмів і даних.

В дисертації розвивається спеціальний математичний апарат, що поєднує засоби АСД і формальної логіки з метою денотаційного представлення функцій, зокрема операторів САА/М. Центральне місце в ньому займає поняття к-інтерпретації, яка являє собою заданий спеціальним чином двомісний предикат, що встановлює області визначення та значень відповідної функції, а також зв'язок між вхідними та вихідними значеннями. Предикат будується таким чином, що значенням його є "істина", коли конфігурації, які в сукупності є аргументами предиката, мають задану об'єктними схемами структуру. Побудова розглядуваного апарату вимагає розширення зображальних засобів АСД. З цією метою вводяться змінні, які дозволяють формулювати складні логічні умови. Запис виду O-c встановлює, що конфігурація, яка розпізнається в момент застосування об'єкта O, позначається змінною застосування об'єкта c. Якщо c - змінна застосування об'єкта O, що зустрічається в тілі циклу, то c(i) позначає конфігурацію, яка розпізнається об'єктом O при i-тій ітерації тіла циклу. Крапкою позначається номер біжучої ітерації, який вважається рівним 0 перед початком виконання циклу та збільшується на 1 після перевірки умови виходу із циклу, якщо значенням умови є "хибність".

Розглянемо запропоноване в дисертації поняття к-інтерпретації функції над конфігураціями, з допомогою якого вводяться та формалізуються поняття (функції), необхідні для специфікації програм в термінах, що відповідають предметній

області розв'язуваної задачі.

Вираз $P(x_1,x_2,\ldots,x_n)$ виду $x_1::O_1,x_2::O_2,\ldots,x_n::O_n$, який приймає логічне значення "істина" (1), якщо для довільного і конфігурація x_i розпізнається об'єктом O_i, та логічне значення "хибність" (0) в супротивному випадку, називається к-предикатом. Змінні застосування об'єктів в к-предикатах вважаються зв'язаними квантором існування. К-предикат

$$K.f(x_1,x_2,\ldots,x_n,y) \equiv f(x_1,x_2,\ldots,x_n)=y$$

називається к-інтерпретацією функції $f(x_1,x_2,\ldots,x_n)$. Функції можна задавати або визначаючи к-інтерпретацію в явному виді, або застосовуючи аксіоматичний підхід та зв'язуючи к-інтерпретації співвідношеннями. В рядках 5-8 таблиці 1 наведено к-інтерпретацію функції сортування числових масивів sort (при цьому використовується функція occmb(x,y), яка встановлює число входжень в масив x деякого елемента y; к-інтерпретацію occmb приведено в дисертації). Змістовно K.sort(x;y) означає наступне: в результаті застосування функції sort до конфігурації x буде отримана конфігурація y, якщо існують конфігурації z, r такі, що x розпізнається об'єктом, заданим формулою, вказаною після x::, та (одночасно) y розпізнається об'єктом, заданим формулою, вказаною після y::.

Логічно пов'язані поняття (функції) та об'єкти об'єднуються в класи - пари виду (O,F), де O - множина об'єктів, F - множина функцій, визначених на об'єктах O. Наприклад, описаний в рядках 2-8 таблиці 1 клас Array задає числові масиви та функції, що застосовуються до них. В термінах функцій класів визначається семантика операторів САА/М, яка формалізується наступним чином.

К-предикатом по множині міток N називається вираз $P_N(x)$ виду $N_1::O_1,\ldots,N_n::O_m$, що приймає логічне значення "істина", якщо в к-конфігурації x наявні конфігурації з мітками $N_i \in N$ та кожна з них розпізнається відповідним об'єктом O_i

(i=1..m), і логічне значення "хибність" в протилежному випадку. К-конфігурація $u=P_{M;N}(x,y)/t$ із множиною міток конфігурацій N називається перетином двомісного к-предиката $P_{M;N}(x,y)$ по к-конфігурації t, якщо $P_{M;N}(t,u)=1$. За означенням, $P_{M;N}(x,y)/t=kundf$, якщо $\forall y: P_{M;N}(t,y)\neq 1$, де kundf позначає невизначену к-конфігурацію.

К-інтерпретацією оператора A (позначається через ${}^{k}A$) називається двомісний к-предикат $P_{M;N}(x,y)$ по множинам міток M і N, такий що:

1) $P_{M;N}(x,y)\ \&\ P_{M;N}(x,z) \to (y=z)$;

2) $\forall s \in KN : A(s)=s\oplus P_{M;N}(x,y)/s$;

де KN - множина к-конфігурацій, $\oplus$ - операція накладення, яка означається так, що $x\oplus y$ являє собою об'єднання к-конфігурацій x і y, із якого вилучено помічені конфігурації, що належать до x, з мітками із перетину множин міток, які використовуються в x та y. Із означення к-інтерпретації випливає, що результатом виконання деякого оператора є накладення перетину його к-інтерпретації на вхідні дані. В рядках 9-12 таблиці 1 наведено к-інтерпретації операторів схеми SORT. (Позначення <c> використовується для об'єкта, що містить лише одну конфігурацію c).

Якщо ${}^{k}A=P_{M;N}(x,y)= M_1::O_1^M,\ldots,M_m::O_m^M;N_1::O_1^N,\ldots,N_n::O_n^N$, то к-предикат $?P_M(x)=M_1::O_1^M,\ldots,M_m::O_m^M$ називається вхідним предикатом оператора A, к-предикат $P_N?(x,y)=N_1::O_1^N,\ldots,N_n::O_n^N$ - вихідним предикатом оператора A, а операторна формула $\hat{A}=[?P]*A*[P?]$ - к-трійкою оператора A, яка є іншою формою зображення к-інтерпретації.

Використовуючи введені поняття, проекти програм пропонується задавати в формі проектних специфікацій - трійок виду <S; {C_i}; {K_j}>, де S - операторна схема, C_i - класи, K_j - к-інтерпретації операторів та умов схеми S, при визначанні яких використано введені в класах C_i функції. При розробці

проектів, що зображаються таким чином, слід описати алгоритм задачі (в формі операторної схеми), оброблювані дані (в формі об'єктних схем), поняття предметної області (в формі функцій класів) та функціональні специфікації (в формі к-інтерпретацій) програми та її складових частин (що ототожнюються з операторами САА/М). Математична строгість застосовуваних при проектуванні засобів забезпечує можливість перевірки (формальної в тому числі) проектів програм. Приклад проектної специфікації наведено в таблиці 1.

Таблиця 1. Проектна специфікація програми сортування

```
1.<SORT=INIT * (Endarr: SHF * INS ) ;
2..( КЛАС Array =
3.    = (eofarr: Real).
4.    sort: Array -> Array
5.     K.sort(x;y) = x::(eofarr: Real-z),
6.       y::(eofarr: Real-r * [(.>1) -> (r(.)> r(.-1))]) *
7.       * [vi>1: ocomb(x,r(i))=ocomb(y,r(i))) &
8.       & ocomb(x,z(i))=ocomb(y,z(i))] )
9. (kSORT= A::Array-x; A::Array-y * [K.sort(x;y)].
10. kINIT= A::Array-x; A::<!> * <x>.
11. kSHF= A::Array-x*(!)*Real-r*Array-y;A::<x>*<r>*<!>*<y>.
12. kINS= A::Array-x*(!)*Real-r*Array-y * [K.sort(x;x)];
          A::Array-z * <!> * <y> * [K.sort(xr;z)] )>
```

Засоби САА/М та АСД, доповнені розглянутими поняттями, утворюють в сукупності понятійну основу мови формалізованих специфікацій і проектування (далі мова САА-специфікацій). В силу особливостей запропонованого математичного апарату ця мова є імперативно-декларативною, наділена розвинутими механізмами структуризації і декомпозиції, поставляє засоби для побудови модельних та аксіоматичних специфікацій, включає в себе незначне число базових понять і позначень, що спрощує її вивчення.

В третьому розділі пропонуються методи обґрунтування правильності проектних специфікацій: тестування, верифікація, трасування. Формалізується поняття правильності проекту, розробляються та досліджуються формальна система, призначена для

доведення правильності проектів програм, та стратегії верифікації, застосування яких ілюструється прикладами.

Тестування проектних специфікацій забезпечується формалізацією поняття виконання оператора САА/М. Подаючи на вхід операторної схеми R к-конфігурацію s та застосовуючи до неї оператори схеми у відповідності з їх к-інтерпретаціями і в порядку, що визначається логікою схеми, на виході отримаємо к- конфігурацію t. Із означення к-інтерпретації випливає, що в проектній специфікації помилки відсутні, коли виконується умова $s\circ^k R/s=t$. Тестування проектних специфікацій є багаторівневим, оскільки проводиться окремо та незалежно для кожної рівності операторної схеми.

Назвемо узгодженою операторну рівність, к-інтерпретації лівої та правої частин якої рівні. Операторна схема узгоджена, якщо узгоджені всі її рівності. Правильність проектів ототожнюється із узгодженістю операторних схем. В результаті задача обгрунтування правильності проектів зводиться до побудови та порівняння к-інтерпретацій.

Таблиця 2. Правила виводу числення Σ

CR: $$\frac{F^*\hat{A}^*\hat{B}^*G,\ R/s={}^kA/s\circ{}^kB/(s\circ{}^kA/s)}{F^*[?R](A^*B)[R?]^*G}$$

DR: $$\frac{F^*(\alpha:\ \hat{A}\backslash\hat{B})^*G,\ R/s::(\alpha:\ \langle{}^kA/s\rangle\backslash\langle{}^kB/s\rangle)}{F^*[?R](\alpha:\ A\backslash B)[R?]^*G}$$

IR: $$\frac{F^*\{\alpha:\ \hat{A}\}^*G,\ R(x,y)=(\exists i>0:\ {}^k(([\neg\alpha]^*A)^i{}^*[\alpha])(x,y))}{F^*[?R]\{\alpha:\ A\}[R?]^*G}$$

s – довільна к-конфігурація; R – к-предикат по множині міток; A, B – довільні, E– тотожній, N – невизначений оператори; α – умова; F,S– формула числення або тотожній оператор ; A^i– композиція із і операторів A.

Для проведення формального обгрунтування правильності проектів (верифікації) пропонується числення к-інтерпретованих схем Σ. Термами в Σ вважаються формули САА/М; правильно побудованими формулами - к-інтерпретовані формули САА/М, в яких

на місці операторів стоять їх к-трійки; аксіомами - к-трійки операторів досліджуваної схеми; правила виводу (таблиця 2) визначають к-інтерпретації операцій САА/М.

Доведено наступні теореми, що характеризують числення $\mathfrak{L}$.

Нехай далі F - операторна формула, $\tilde{F}$ - к-інтерпретована формула, що відповідає F, R - к-предикат.

ТЕОРЕМА 1 (про несуперечність числення). Якщо в численні $\mathfrak{L}$ із $\tilde{F}$ виведено к-трійки [?P]F[P?] та [?R]F[R?], то P=R.

ТЕОРЕМА 2. Якщо в численні $\mathfrak{L}$ із $\tilde{F}$ виведено формулу [?R]F[R?], то R - к-інтерпретація F.

Операторна формула належить до класу $\mathfrak{K}$ операторних формул, що к-інтерпретуються, якщо для довільної її підформули існує к-інтерпретація.

ТЕОРЕМА 3 (про повноту числення). Якщо F належить до класу $\mathfrak{K}$, R - к-інтерпретація F, то в численні $\mathfrak{L}$ із $\tilde{F}$ виводиться формула [?R]F[R?].

Із теорем 2 і 3 та означення узгодженості операторних рівностей випливає теорема 4.

ТЕОРЕМА 4. Операторна рівність A=F, де F належить до класу $\mathfrak{K}$, узгоджена тоді і лише тоді, коли в численні $\mathfrak{L}$ із $\tilde{F}$ виводиться формула $[?^{k}A]F[^{k}A?]$.

При згорточній стратегії доведення узгодженості із формули вибирається підформула, до якої застосовується придатне правило виводу і отримувана в результаті к-трійка підставляється в формулу замість цієї підформули. Наприклад, застосування правила виводу CR до композиції $\hat{SHF}*\hat{INS}$ в схемі SORT дає к-трійку:

[A::Array-x*<!>*Real-r*Array-y * [K.sort(x;x)]] (SHF*INS)
[A::Array-z * <!> * <y> * [K.sort(xr;z)]],

яка заміщує в формулі підформулу $\hat{SHF}*\hat{INS}$. Рівність узгоджена, якщо із формули її правої частини виводиться к-трійка, що відповідає к-інтерпретації лівої частини рівності. В залежно-

сті від напрямку сканування формули, в процесі якого застосовується будь-яке з правил виводу, як тільки це стане можливим, розрізняються лівостороння, правостороння та змішана - ліво- і правостороння почергова або одночасна (паралельна) - стратегії верифікації.

Розгорточна стратегія застосовується, якщо формулу можна представити у вигляді композиції A*B так, щоб по відомим $^{k}_{(A*B)}$ і $^{k}_{A}$ визначити $^{k}_{B}$. Якщо B оператор, то його обчислена к-інтерпретація порівнюється із заданою; в протилежному випадку "розгортається" B: формується композиція B=C*D із відомою к-інтерпретацією $^{k}_{C}$, по $^{k}_{B}$ і $^{k}_{C}$ знаходиться $^{k}_{D}$ і т.д. Доведення вважається успішним, якщо всі к-інтерпретації, обчислені в процесі розгортки, співпадають із заданими. Розгорточна стратегія корисна в поєднанні зі згорткою.

Трасування проектних специфікацій близьке в ідейному плані до символічного виконання програм. У відповідності з деяким критерієм обгрунтування в операторній схемі вибирається траса - послідовність операторів, отримувана в результаті призначення певних значень умовам схеми. Будується к-інтерпретація траси. К-інтерпретація схеми R визначає к-інтерпретацію довільної траси T, тому повинна виконуватись імплікація $^{k}_{T} \to {}^{k}_{R}$, - в протилежному випадку в проекті містяться помилки.

В четвертому розділі на основі запропонованого математичного апарату розробляється метод доказового багаторівневого структурного проектування програм (БСПП/Д) та підтримуючий його інструментарій (система САА/ПРОЦЕСИСТ). Обговорюються сфери застосування та перспективи розвитку розглянутого підходу.

Метод БСПП/Д базується на тій же ідеології розробки ПО, що і метод БСПП (багаторівневе проектування схеми програми засобами САА/М, програмна реалізація базових операторів та умов, генерація програми за схемою та програмними фрагментами, що

відповідають базовим операторам та умовам). Поряд з цим розширено область життєвого циклу ПЗ, яка охоплюється методом: функціональна специфікація, верифікація проектних специфікацій, тестування і відладка; систематично застосовуються як неформалізовані, так і формалізовані специфікації; з метою проведення доказового проектування застосовується багаторівнева модель життєвого циклу ПЗ. Розробка програм за методом БСПП/Д проходить через три рівні, що послідовно конкретизуються. Рівні відрізняються способом інтерпретації операторів, умов, функцій (далі примітивів) проектної специфікації (таблиця 3).

Таблиця 3. Розробка програм за методом БСПП/Д

1-й рівень

.*постановка задачі, проектування алгоритму і даних* - *розробка об'єктних і операторних схем*, класів;
.*розробка т-інтерпретацій* (стилізованих описів на звичайній мові) примітивів проектної специфікації;
.узгодження: тестування, *трасування*, верифікація т-інтерпретованих схем.

2-й рівень

.*розробка к-інтерпретацій* примітивів проектної специфікації на основі їх т-інтерпретацій;
.узгодження: тестування, *трасування*, верифікація к-інтерпретованих схем

3-й рівень

.*розробка п-інтерпретацій*- програмних реалізацій примітивів проектної специфікації - на основі їх т- і к-інтерпретацій;
.узгодження: відладка згенерованої за описами програми - *пооператорне* (в розумінні САА/М) *виконання програми, згенерованої за операторною схемою і заданими реалізаціями операторів, на тестових наборах даних.*

На кожному з рівнів перевіряється узгодженість рівностей, виявляються та усуваються помилки. Для перевірки узгодженості т- і к-інтерпретованих рівностей застосовуються тестування, трасування, верифікація. Однак в першому випадку предикати задаються на звичайній мові. Узгодження п-інтерпретованих схем означає відладку програм. Відповідність між п- і к-інтерпретаціями може бути строго доведена. Для цього

придатні традиційні методи верифікації програм.

Характерними ознаками системи САА/ПРОЦЕСИСТ, що являє собою результат розвитку системи МУЛЬТИПРОЦЕСИСТ, є: трирівнева модель життєвого циклу ПЗ, на кожному з яких виконується процедура обгрунтування узгодженості; діалоговий інтерфейс; застосування синтаксично-орієнтованих редакторів для вводу схем та інтерпретацій; застосування двох мов (підмножин вхідної мови системи МУЛЬТИПРОЦЕСИСТ) для предоставлення операторних схем; розвинута довідкова служба. Підтримувані системою САА/-ПРОЦЕСИСТ етапи розробки програм за методом БСПП/Д виділено в таблиці 3 курсивом. В даній версії система САА/ПРОЦЕСИСТ орієнтована на створення програм на мові REXX в ПДО СВМ. Система включає в себе 37 програм, написаних на мовах REXX та EXEC2 (загальний об'єм близько 5 тис. рядків), використовує редактор XEDIT ПДО СВМ та засоби управління екраном дисплея пакета DMS CMS.

В додатку 1 наведено доведення властивостей числення к-інтерпретованих схем. В додатку 2 міститься проектна специфікація системи САА/ПРОЦЕСИСТ.

ВИСНОВКИ

Наступні результати, отримані в дисертації, виносяться на захист.

1. Побудовано логіко-алгебраїчну модель, орієнтовану на створення багаторівневих проектів програм, на основі якої:
- із застосуванням введених понять (к-предиката, перетину к-предиката, класу та ін.) і сформульованої концепції к-інтерпретації, розвинуто підхід до розробки проектів програм та визначення семантики абстрактних перетворювачів даних, що використовуються при проектуванні;
- запропоновано формалізм проектної специфікації як засіб подання проектів програм;

- формалізовано семантику виконання оператора алгебри алгоритмів, поняття правильності проекту програми та розроблено методи аналізу проектних специфікацій: тестування, верифікація, трасування (символічне виконання).

2. Побудовано асоційоване із запропонованою моделлю числення к-інтерпретованих операторних схем, аксіомами в якому вважаються к-інтерпретації (функціональні специфікації) операторів алгебри алгоритмів, а правила виводу характеризують властивості операцій алгебри алгоритмів. Доведено теореми про повноту та несуперечність числення і показано, що доведення правильності проектів програм зводиться до виводу формул числення. Запропоновано стратегії проведення формального доведення: згорточну, розгорточну, змішану (в тому числі паралельну).

3. На основі створеного математичного апарату розроблено метод багаторівневого доказового структурного проектування програм (БСПП/Д), в якому використовується багаторівнева модель життєвого циклу ПЗ і на кожному з рівнів проводиться перевірка проектів розроблюваних програм із застосуванням запропонованих методів тестування, верифікації та трасування проектних специфікацій.

4. В рамках створення програмного забезпечення методу БСПП/Д:
- визначено понятійну основу мови (САА-специфікацій) специфікації і проектування програм декларативно-імперативного типу з розвинутими зображальними можливостями для подання модельних і аксіоматичних специфікацій;
- розроблено і оформлено засобами мови САА-специфікацій проектну специфікацію інструментарію БСПП/Д.

5. В рамках розвитку РИТМ-технології, що проводиться у відповідності з Білоруською республіканською комплексною науково-технічною програмою "Інформатика" (номер держ. реєстрації 01.90.0 036397), реалізовано інструментарій, який підтримує основні етапи розробки програм за методом БСПП/Д.

Основні положення дисертації опубліковані в роботах:

1. Бакулин А.В., Грицай В.П., Моравский Р.Р. Организация диалога в системе МУЛЬТИПРОЦЕССИСТ // Организация взаимодействия человека с ЭВМ : Сб. научных трудов. - Киев: ИК АН УССР, 1985.- С. 22-23.

2. Бакулин А.В. О реализации языка управления системой МУЛЬТИПРОЦЕССИСТ // 7-я Всесоюз. шк.-семинар "Параллельное программирование и высокопроизводительные системы": Тез. докл.- Киев: ИК АН УССР, 1986.- С. 52-53.

3. Костырко В.С., Бакулин А.В. Об индуктивном синтезе инвариантных утверждений и функций программ // Кибернетика.- 1986.-N1.- С.18-24.

4. Бакулин А.В. Об организации архива системы МУЛЬТИПРОЦЕССИСТ // 8-й Всесоюзный семинар "Параллельное программирование и высокопроизводительные системы": Тез. докл.- Киев: ИК АН УССР, 1988.- С. 56-57.

5. Бакулин А.В., Цейтлин Г.Е. Об одном подходе к алгебраической спецификации программ // 2-я Всесоюз. конф. по прикладной логике. Новосибирск, 7-9 июня 1988 г.: Тез. докл.- Новосибирск, 1988.- с. 234-236.

6. Бакулин А.В. Об одном подходе к спецификации и проектированию программ // 10-я Всесоюз. шк.-семинар "Параллельное программирование и высокопроизводительные системы": Тез. докл.- Киев: ИК АН УССР, 1990.- С. 61-62.

7. Бакулин А.В. Математическая модель языка спецификаций и проектирования программ.- Мн., 1991.- 25с.- (Препринт /АН БССР. Вычислительный центр; N 4(4)).

8. Бакулин А.В. Многоуровневая разработка программ, основанная на применении интерпретированных схем алгебры алгоритмов .- Мн., 1991.- 23с.- (Препринт /АН БССР. Вычислительный центр; N 5(5)).

9. Цейтлин Г.Е., Бакулин А.В. Многоуровневые структурированные проекты программ и их обоснование // Кибернетика и системный анализ.- 1991.- N5.- С. 98-107.

Підписано до друку 20.11.92. Формат 60x84/16.
Умов.друк.арк. 1,16. Умов.фарб.-відб. 1,28. Обл.-вид.арк. 0,85.
Тираж 45 экз. Заказ 91 . Безкоштовно.

Обчислювальний центр АН Білорусі.
220072, Мінськ, вул. Ф.Скорини, 25.
Надруковано на ротапринті Обчислювального центру АН Білорусі.
220072, Мінськ, вул. Сурганова, 11.