де $C = max\{c_1, c_2, \frac{1}{c_2}\}$; M – константа, яка не залежить від τ, h та u.

1. Гордезиани Д.Г., Самарский А.А. Некоторые задачи термоупругости пластин и оболочек и метод суммарной аппроксимации // Комплексный анализ и его приложения. М., 1978. С. 173-186. 2. Кузык А.М., Макаров В.Л. Скорость сходимости метода суммарной аппроксимации для обобщенных решений // Докл. АН СССР 1984. Т. 275. № 2. С. 297-301. 3. Макаров В.Л., Кузык А.М. Сходимость метода суммарной аппроксимации для обобщенных решений. Львов, 1984. 34 с. Рукопись деп. в УкрНИИНТИ, № 2096 Ук-Д84. 4. Самарский А.А. Теория разностных схем. М., 1983.

Стаття надійшла до редколегії 26.02.86

УДК 681.3.06

О.О.Євтушенко, В.В.Черняхівський

ДО ВЕРИФІКАЦІЇ ЦИКЛІВ ТИПУ FOR

Одержання якісного програмного продукту – основна мета системи КАПРИ [1]. Ряд її підсистем з різних позицій підходить до розв"язання цієї проблеми. Наприклад, згенеровані користувачем у діалозі з КАПРИ і структуровані ФОРТРАН-програми можуть налагоджуватися системою КАПКАН методом пошуку конкретних помилок [2], системою тестування, а також системою СИНВЕР (система інтерактивної верифікації). У результаті (після обробки кожною з указаних систем) якість налагодженої програми неодмінно повинна підвищитись.

Кожна програма, якщо вона нормально завершується, сама по собі не будучи ні правильною, ні неправильною, виконує деяку функцію у широкому розумінні цього слова. Визначення відповідності вказаної функції і тієї, реалізацію якої передбачає технічне завдання на дану програму, стає можливим при роботі користувача з інтерактивною системою верифікації СИНВЕР. При розробці цієї системи враховували спеціальну архітектуру поступаючих на її вхід програм, кожна з яких складена на підмножині мови ФОРТРАН ОС ЕС згідно з рекомендаціями [3]. Відзначимо, що СИНВЕР орієнтується на дослідження програм, які реалізують обчислювальні задачі. Суттєво використовується і те, що доведення коректності ієрархічно збудованої програми можна розкласти на доведення коректності елементарних програм, розміщених на відповідних рівнях ієрархії.

Цілий ряд обчислювальних ФОРТРАН-програм, як показує практика, має досить примітивну структуру, включаючи використання тільки таких програмних конструкцій, як послідовність, розвилка і частковий випадок циклу while – ітеративний цикл типу for (для ФОРТРАНа – цикл DO). У зв"язку з цим для деяких комбінацій таких програмних структур у системі СИНВЕР реалізовані алгоритми автоматичної генерації відповідних програмних функцій. При цьому, якщо X – деякий початковий стан даних програми P , а Y – відповідний вихідний стан даних програми P після нормального завершення її роботи, то множина всіх можливих впорядкованих пар $\{(X,Y)\}$ визначає програмну функцію P . Для побудови програмної функції ділянок програми вказаного виду від програміста не вимагається задання жодних специфікацій. Він виконує роль контролера, визначаючи в діалозі відповідність інформації, яку видала система, технічному завданню.

Розглянемо більш детально завдання генерації програмної функції для ітеративного циклу, керованого оператором ФОРТРАНа DO ; тіло циклу – арифметичні оператори присвоювання, що означає розміщення в структурованій програмі на найнижчому рівні ієрархії. Нехай вказаний цикл має такий вигляд:

```
      DO  X  I = α1, α2, α3
          q1
          ⋮
          qk
 X    CONTINUE,
```

(I)

де $q_1, \ldots, q_k$ – арифметичні оператори присвоювання. Введемо ряд понять. Вектором стану даних $\vec{X}$ програми P назвемо вектор, компонентами якого є всі ідентифікатори змінних, що містяться у тексті P . Область допустимих значень кожної змінної за припущенням – множина або дійсних, або цілих, або комплексних машинних чисел. Розширимо цю область елементом ω , що означає "не визначено". Тоді початковим вектором стану даних програми P назвемо вектор $\vec{X^0}$, який отримується з $\vec{X}$ відповідною заміною всіх ідентифікаторів змінних на значення цих змінних до початку роботи програми, а вихідним вектором стану даних Y вектор, побудований аналогічно зі значень цих змінних після нормального завершення програми. У результаті програмна функція $[P]$

програми P визначається як

$$[P] = \{(\vec{X}^o, \vec{Y})\}.$$

Кожний арифметичний оператор присвоювання можна вважати деякою програмою, яка перетворює початковий вектор стану даних у вихідний. Тобто, якщо програма P складається з одного оператора присвоювання φ, то $\varphi = \{(\vec{X}^o, \vec{Y})\}$ і $[P] = \{(\vec{X}^o, \vec{Y}) | Y = \varphi(\vec{X}^o)\}$. Розглянемо послідовність операторів

$$q_1, \dots, q_K. \tag{2}$$

Вони відповідно мають вигляд: $p_i = f_i(\eta_1^i, \dots, \eta_{z_i}^i), (1 \le i \le K)$, де $p_i, \eta_j^i, (1 \le i \le K, 1 \le j \le z_i)$ – ідентифікатори змінних; $f_i, (i = \overline{1, K})$ – позначення для правих частин операторів присвоювання, що містять ці змінні. Саме $f_1, \dots, f_K$ змінюють стан даних програми (2). Тому $[P] = \{(\vec{X}^o, \vec{Y}) | \vec{Y} = f_K(f_{K-1}(\dots f_1(\vec{X}^o)\dots))\}$, де P – програма (2). Для дослідження коректності такої програми розглянемо ті компоненти $\vec{X}^o$, які після її завершення відрізняються від відповідних компонент Y. Очевидно, що таких компонент може бути d, де $1 \le d \le K$ і $d = K$, якщо $p_1 \ne p_2 \ne \dots \ne p_K$. Позначимо множину змінних, що знаходяться в лівих частинах операторів присвоювання, через M:

$$M = \{p_{l_1}, \dots, p_{l_d}\}, (1 \le l_i \le K, 1 \le i \le d). \tag{3}$$

Змінні (3) назвемо частковими результуючими змінними (ЧРЗ) послідовності операторів (2).

Нехай $p_j \in M$. Тоді зрозуміло, що існує така підпослідовність послідовності (2) (навіть коли вона складається з одного елемента) $q_{t_1}, q_{t_2}, \dots, q_{t_s}$, що $p_j = p_{t_1} = p_{t_2} = \dots = p_{t_s}, t_1 < t_2 < \dots < t_s, (1 \le s \le K)$. Тоді після виконання оператора q_{t_s} значення змінної p_j не зміниться, тобто q_{t_s} формує вихідне значення ЧРЗ p_j. Назвемо оператор q_{t_s} послідовності (3) частковим результуючим оператором (ЧРО) часткової результуючої змінної p_j, якщо жоден із операторів (2) з порядковим номером, більшим за t_s, не містить у лівій частині p_j. Часткову результуючу змінну p_j назовемо результуючою змінною (РЗ) послідовності операторів $q_1, \dots, q_K$, якщо ЧРО змінної p_j – це оператор q_K або q_{j_0} – ЧРО змінної $p_j \in M$, де $(1 \le j_0 \le K)$ і $(\forall l)$ $(j_0 < l \le K)$, q_l не містить у правій частині ідентифікатор p_j. Результуючим оператором (РО) послідовності (2) назвемо оператор, порядковий номер якого найбільший серед тих, що містять РЗ p_j.

Нехай $\vec{X}=(\xi_1,\ldots,\xi_m)$, $\vec{X}^0=(\xi_1^0,\ldots,\xi_m^0)$, $\vec{Y}=(\xi_1^*,\ldots,\xi_m^*)$ відповідно вектор стану даних програми, початковий вектор і вихідний вектор стану даних програми (I). Припустимо, що (I) працює нормально, тобто обов"язково завершується, виконавши тіло циклу NN разів. Тоді, замінивши (I) послідовністю

$$\underbrace{q_1,\ldots,q_K}_{1\text{-й РАЗ}},\ \underbrace{q_1,\ldots,q_K}_{2\text{-й РАЗ}},\ldots,\underbrace{q_1,\ldots,q_K}_{NN\text{-й РАЗ}}, \qquad (4)$$

вихідний вектор програмної функції циклу визначаємо за формулою:

$$\underbrace{q_K\circ q_{K-1}\circ\cdots q_1\circ}_{NN\text{-й РАЗ}}\ldots\circ\underbrace{q_K\circ\ldots\circ q_1\circ}_{2\text{-й РАЗ}}\underbrace{q_K\circ\ldots\circ q_1}_{1\text{-й РАЗ}}(\vec{X}^0)=\vec{Y}.$$

Зрозуміло, що (4) має ті ж ЧРЗ, що і (2). Тому очевидно, що цикл змінює у процесі роботи значення ЧРЗ послідовності (2) і тільки їх (не беручи до уваги (I)), тобто

$$\xi_j^*=\xi_j^0, \qquad (5)$$

якщо ξ_j не є ЧРЗ (2). Нехай без зменшення загальності компоненти вектора $\vec{X}$, $\xi_1,\ldots,\xi_d$ – ідентифікатори змінних, що утворюють множину M. Введемо d функцій:

$$(\forall i)(1\leq i\leq d)\ \varphi_i: U_i\to V_i,\ \varphi_i(\xi_i,\vec{X}^0)=\xi_i^*, \qquad (6)$$

де U_i - множина впорядкованих пар виду $(\xi_i,\vec{X}^0)$, $\xi_i\in\vec{X}$;
V_i - множина допустимих значень змінної з ідентифікатором ξ_i. Тоді, якщо відомі початкові значення компонент вектора $\vec{X}^0$ $\xi_1^0,\ldots,\xi_m^0$ і функції φ_i, то значення компонент вектора $\vec{Y}$ для програмної функції циклу (I) визначається за формулами (5)-(6).

Для автоматичної генерації аналітичного виду функцій φ_i в системі СИНВЕР розроблені алгоритми, які використовують символічне виконання, семантичний аналіз ЧРЗ послідовності (2), а також (в окремих випадках) метод математичної індукції по NN - числу раз виконання циклу (I).

I. Архангельский Б.В. Разработка промышленных методов и систем создания качественных программных изделий. Проект

КАПРИ-I // Надежность и качество программного обеспечения: Тез. докл. Всесоюз. конф. Львов, 27-29 января 1985 г. С.9-II. 2. Архангельский Б.В., Черняховский В.В. КАШКАН-ФОРТРАН-система отладки программ методом поиска конкретных ошибок // Программирование. 1984. № 6. С. 30-40. 3. Дзержинский Ф.Я. ПСЕВДОКОД (язык проектирования программ и правила структурного программирования). М., 1979.

Стаття надійшла до редколегії 03.02.86

УДК 519.34:532.516

В.М.Зубов, Г.А.Шинкаренко

ЗБІЖНІСТЬ МЕТОДУ РЕГУЛЯРИЗАЦІЇ ДЛЯ КРАЙОВОЇ ЗАДАЧІ З РІВНЯННЯМИ НАВ"Є-СТОКСА

Узагальнимо оцінки збіжності методу регуляризації для задачі Стокса, отриманні у праці [1], на випадок стаціонарних рівнянь Нав"є-Стокса.

1. Розглянемо крайову задачу для системи рівнянь Нав"є-Стокса в обмеженій області $\Omega \in R^n$ (n = 2 або 3) з кусково-неперервною границею Γ :

$$\rho(\bar{u}\bar{\nabla})\bar{u} - \mu\Delta\bar{u} + grad\, p = \rho\bar{f} \quad \text{в } \Omega, \tag{1}$$

$$div\,\bar{u} = 0 \quad \text{в } \Omega, \tag{2}$$

$$\bar{u} = 0 \quad \text{на } \Gamma. \tag{3}$$

Ця задача, як відомо, описує в області Ω рух в"язкої нестисливої рідини з постійними в"язкістю $\mu > 0$ і густиною $\rho > 0$ під дією масових сил, заданих вектором $\bar{f} = (f_1, \dots, f_n)$. Треба знайти вектор швидкостей рідини $\bar{u} = (u_1, \dots, u_n)$, а також тиск p , які б задовольняли рівняння руху (1), умову нестисливості (2) і однорідну крайову умову Діріхле (3).

2. У варіаційній постановці задача (1)-(3) має наступний вигляд [3]:

$$a(\bar{u},\bar{v}) - b(p,\bar{v}) + a_1(\bar{u},\bar{u},\bar{v}) = \langle \bar{f},\bar{v} \rangle \quad \forall \bar{v} \in V, \tag{4}$$

$$b(q,\bar{u}) = 0 \qquad \forall q \in Q, \tag{5}$$