МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ ДНР

ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Институт КНТ

Факультет ФИСП

Кафедра ПИ им. Л.П. Фельдмана

Лабораторная работа №5

Тема: «Доказательство правильности программ методом программных функций»

Курс: Качество программного обеспечения и тестирование

Выполнил

ст. гр. ПИ-18Б

Моргунов А.Г.

Проверил

Федяев О. И.

Ищенко А. П.

Донецк – 2021

# Задание (Вариант 5)

Методом программных функций доказать правильность программы, выполняющей заданную обработку массива (см. табл.1). Составить структурированную программу и выполнить ее верификацию с использованием программных функций.

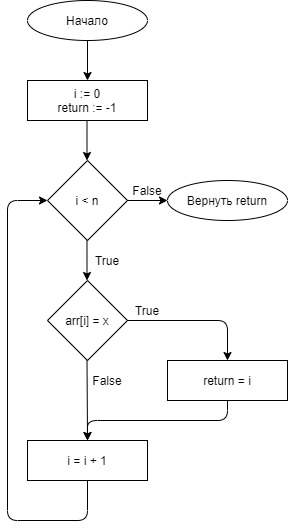
Определение места заданного элемента в массиве.

Исходные данные:

1. x – искомый элемент

2. arr – исходный массив

3. n – размер исходного массива



Определим дополнительную функцию FND, которая будет искать заданный элемент в массиве, и возвращать его индекс.

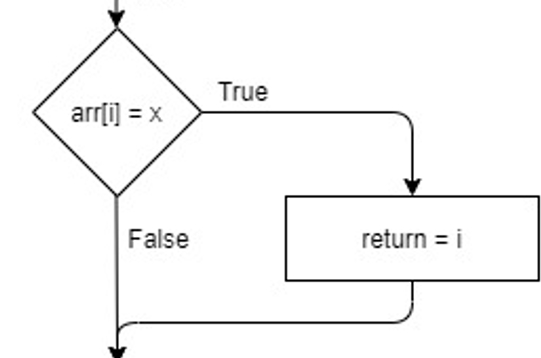
Сформулируем гипотезу о программной функции:

F = (n>=1) -> (return, i := FND(x, arr[0:n-1], n-1)

Докажем, что F = [P].

Выделим элементарные программы в исходной программе.

**Элементарная программа P1**



Сформируем гипотезу о ее функции

F1 = (return := FND(x, arr[i])

Трассировочные таблицы для разных путей.

Путь 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Оператор | Предикат | Arr[i] | Return |
| 1 | Arr[i] = x | (Arr[i]=x)=True | Arr[i]0 | Return0 = -1 |
| 2 | Return := i |  | Arr[i]1 = Arr[i]0 | Return1 = i |

Предикат пути:

Arr[i]=x

Данные:

Arr[i]1 = Arr[i]0

Return1 = i

Функция:

(Arr = x) -> (return := i)

Путь 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Оператор | Предикат | Arr[i] | Return |
| 1 | Arr[i] ≠ x | (Arr[i]=x)=False | Arr[i]1 = Arr[i]0 | Return1 = -1 |

Предикат пути:

Arr[i]≠x

Данные:

Arr[i]0

Return1 = -1

Функция:

(Arr ≠ x) -> (return := -1)

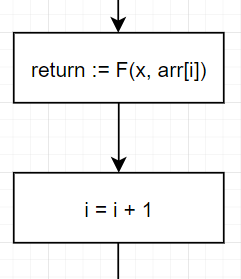
После объединения функций первого и второго пути выполнения программы получаем следующую функцию:

[P1’] = (Arr = x) -> (return := i) | (Arr ≠ x) -> (return := -1)

Отсюда следует правильность программы P1

F1 = P1 = [P1’]

**Элементарная программа P2**



Сформируем гипотезу о ее функции

F2 = (return, i := FND(x, arr[i]), i + 1)

Трассировочные таблицы для разных путей.

Путь 1

|  |  |  |  |
| --- | --- | --- | --- |
| № | Оператор | Return | i |
| 1 | return = FND(x, arr[i]) | Return1 = FND(x, arr[i]) | I1 = I0 |
| 2 | i := I + 1 | Return2 = Return1 | I2 = I1 + 1 |

Данные:

Return1 = Return0 = FND(x, arr[i])

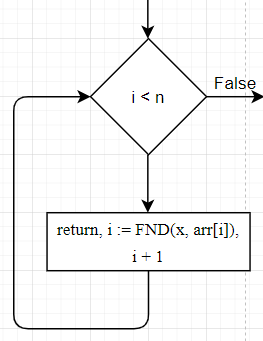
I2 = I0 + 1

Функция:

[P2] = (return, i := FND(x, arr[i]), i + 1)

P2 совпало с гипотезой, что означает правильность программы.

**Элементарная программа P3**



Сформируем гипотезу о ее функции

F3 = (I < n) -> (return, I := FND(x,arr[i:n-1]), n-1)

Так как рассматриваемый фрагмент представляет собой цикл, то нужно преобразовать Р3 в рекурсивную программу P3'.

Применим лемму о сведении циклической программы к рекурсивной P3':

**if** i<n **then** return, i := FND(x,arr[i:n-1]), i+1;

(i <n) → return, i := FND(x,arr[i:n-1]), n-1

**fi**

По лемме следует, что [P3] = [P3'].

Найдём программную функцию [P3'].

Опять воспользуемся трассировочными таблицами для каждого пути выполнения программы P3'

Путь 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Оператор | Предикат | return | i |
| 1 | i<n | (i0<n)=True | Return1 = Return0 | i1= i0 |
| 2 | return, i = FND(x, arr[i]), i+1 |  | Return2 = FND(x, arr[i1]) | i2= i1 + 1 |
| 3 | i<n | (i3<n)=True | Return3 = Return2 | i3= i2 |
| 4 | return, i = FND(x, arr[i]), i+1 |  | Return4 = FND(x,arr[i3:n-1]) | i4= n-1o |

**Предикат пути:**

(i0<n)&( i3<n) = (i0<n)&( i0+1<n) = (i0<n-1) = True

**Данные**:

Return1 = FND(x, arr[i0])

Return4 = FND(x,arr[i3:n-1])

Если объединить данные, то получим:

Return4 = FND(x,arr[i0:n-1])

I5 = n

**Функция:**

(I < n-1) -> (return, I := FND(x,arr[i:n-1]), n-1)

Путь 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Оператор | Предикат | return | i |
| 1 | i<n | (i0<n)=True | Return1 = Return0 | i1= i0 |
| 2 | return, i = FND(x, arr[i]), i+1 |  | Return2 = FND(x, arr[i1]) | i2= i1+1 |
| 3 | i<n | (i3<n)=False | Return3 = Return2 | i3= i2 |

**Предикат пути:**

(i0<n)&( i3>=n) = (i0<n)&( i0+1>=n) = (i0 = n -1) = True

**Данные:**

Return1 = FND(x, arr[i0])

i3= i2= i1+1= i0 + 1

**Функция:**

(I = n -1) -> (return, I := FND(x,arr[i]), n-1)

Так как в данном случае i = n - 1, то функцию можно записать в виде

(I = n-1) -> (return, I := FND(x,arr[i:n-1]), n-1)

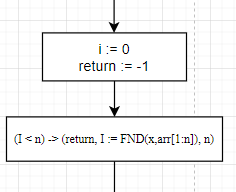
После объединения функций на первом и втором путях выполнения программы получаем следующую функцию:

[P3’] = (I < n) -> (return, I := FND(x,arr[i:n-1]), n-1)

Отсюда следует правильность программы P3

F3 = P3 = [P3’]

**Элементарная программа P4**



Она получена путём замещения цикла его программной функцией (на основании аксиомы замещения).

Гипотеза для этой программы естественно должна совпадать с основной функцией.

F4 = (n>=1) -> (return, i := FND(x, arr[0:n-1]), n-1)

Докажем, что f4= [P4]

Трассировочные таблицы для разных путей.

Найдём программную функцию программы Р4.

[Р4] = ? Используем трассировочную таблицу.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Оператор | Предикат | Return | i |
| 1 | i:=0 |  | Return1 = Return0 | I1=0 |
| 2 | (I < n) -> (return, I := FND(x,arr[0:n-1]), n-1) | (i1<n)=True | Return2 = FND(x, arr[i1]) | I1=n-1 |

**Предикат пути:**

(i1<n) = i0 <n = 0 < n

**Данные:**

Return1 = FND(x, arr[i0])

I1=n -1

**Функция**:

[P4] = (n>=1) -> (return, i := FND(x, arr[0:n-1]), n-1)

Видно, что F4= [P4] , значит доказано равенство F = [P] .