ЗВІТ ПРО ВИКОНАНЕ ЗАВДАННЯ

по курсу "Методи верифікації і оптимізації програм"
студента групи ПК-16м-1
Бекленищева Владислава Ігоровича

ЗАВДАННЯ № 1

кафедра комп'ютерних технологій, ДНУ
2016/2017 навч. р.

Постановка задачи

Вариант - 1

Даны натуральные числа: N, a[1], a[2], ..., a[N]. Определить количество членов a[K] последовательности a[1], ..., a[N], имеющих четные порядковые номера и являющихся нечетными числами.

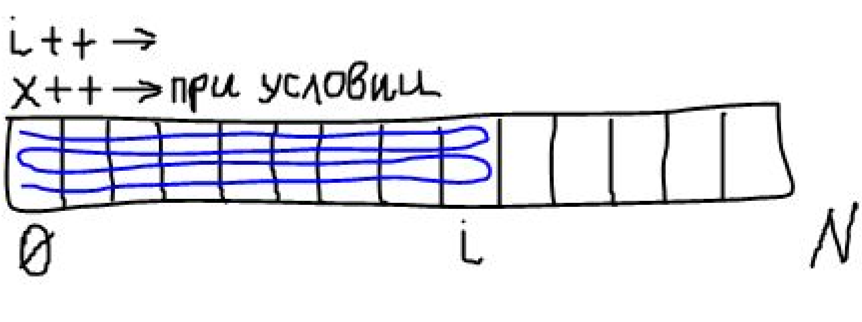
Описание решения задачи

Основная идея решения задачи состоит в том чтобы пройтись по всем элементам массива от начала до конца и подсчитать количество элементов, которые стоят на четных порядковых номерах (even) и одновременно являются нечетными числами (odd). Проверку на четность / нечетность можно выполнять с помощью операции “взятии остатка от деления” (%), однако в доказательстве вместо выражений с этой операцией используются соответствующие предикаты even и odd. Например, следующий образом проверяется нечетность элемента, стоящего на позиции i: $odd(a[i])$.

Основной стартовой точкой для доказательство является построение спецификации программы. Так как программа является циклом с предварительной инициализацией, кроме предикатов предусловия Q, программы S и предиката постусловия R, нужно еще составить предикаты I, R и t, которые являются командой инициализации, инвариантом и ограничивающей функцией соответственно.

Для удобства предположим, что индексация массива производится от 0 до N-1, а N может равняться или быть больше 0 (например, когда массив пустой). Сразу отсюда следует предусловие Q: $\{N \geq 0\}$.

Далее нужно найти инвариант. Инвариант - предикат, истинность которого не должна нарушаться в ходе работы цикла и перед его началом. Это понятие является довольно важным в теории верификации программ и самом программировании, так как при разработке цикла нужно как раз реализовать его инвариант. В данной задачи, инвариант может выглядеть следующим образом:



На рисунке синим отмечена область, которую мы уже прошли в процессе поиска. Верхняя граница синей области имеет значение i. Таким образом, в данной задаче инвариант состоит из двух условий:

- Указатель на элемент нужно двигать от начала массива (от 0) в конец (до N), передвигая его на 1 элемент вперед на каждой итерации (i ++) .
- При нахождении нечетного элемента, стоящего на четном месте (проверяем i), стоит увеличить счетчик количества на 1 (x ++) .

Для нахождения количества элементов в теории доказательства правильности программ есть квантор количества. Таким образом, инвариант в предикатной форме можно подать в таком виде:

$$\{inv\ P:\ 0 \le i \le N \wedge x = (\aleph_j :\ 0 \le j < i :\ even(i) \wedge odd(a[j]))\}$$

С инварианта можно получить постусловие, то есть предикат R. Постусловием является то, что должно быть результатом работы программы. В данной задачи нас интересует количество чисел, которое возвращает квантор количества:

$$\{R:\ x = (\aleph_j :\ 0 \le j < N :\ even(i) \wedge odd(a[j]))\}$$

Кроме инварианта еще одним важным предикатом, является целочисленная неотрицательная, ограничивающая функция t (bound t). Во время работы цикла она должна уменьшаться и приближаться к 0 (нижняя граница). В данной задачи t однозначно зависит от числа шагов, которые осталось выполнить, то есть t будет равна: $\{bound\ t:\ N - i\}$. Таким образом, на каждой итерации t уменьшается в то время, как i увеличивается на 1.

Команда инициализации может быть построена из инварианта P и предусловия Q. В данном случае она будет следующей: $i, x := 0, 0;$ То есть указатель i показывает на первый элемент массива, а x, то есть количество, в самом начале равно 0.

Охрана цикла записывается исходя из ограничивающей функции, то есть N - i > 0. Сама программа S записывается для данной задачи в виде IF конструкции: если элемент на четной позиции и нечетный, то увеличиваем счетчик x и i, а есть условие не выполняется, то увеличиваем только i. Всю команду повторения можно записать в таком виде:

$$\begin{array}{lll} do\ i < N \rightarrow & if\ even(i) \wedge odd(a[i]) & \rightarrow x, i := x + 1, i + 1 \\ & \quad |\neg((even) \wedge odd(a[i])) & \rightarrow i := i + 1 \\ & fi & \\ & & \\ od & & \end{array}$$

Само доказательство (вместе со спецификацией) программы с комментариями описано в следующем пункте отчета.

Спецификация и доказательство программы

$\{N \geq 0\}$
$i, x := 0, 0;$
$\{inv\, P : 0 \leq i \leq N \wedge x = (\aleph_j : 0 \leq j < i : even(i) \wedge odd(a[j]))\}$
$\{bound\, t : N - i\}$

$do\ \ i < N \rightarrow if\ even(i) \wedge odd(a[i]) \quad \rightarrow x, i := x + 1, i + 1$
$\quad\quad |\neg((even) \wedge odd(a[i)) \quad \rightarrow i := i + 1$
$\quad fi$
od

$\{R : x = (\aleph_j : 0 \leq j < N : even(i) \wedge odd(a[j]))\}$

1) $Q \Rightarrow WP(I, P)$

$N \geq 0 \Rightarrow WP(i, x := 0{,}0\, ,\, 0 \leq i \leq N \wedge x = (\aleph_j : 0 \leq j < i : even(i) \wedge odd(a[j]))$
// выполним подстановку:
$N \geq 0 \Rightarrow 0 \leq 0\ \&\ 0 \leq N \wedge 0 = (\aleph_j : 0 \leq j < i : even(i) \wedge odd(a[j]))$
// $0 \leq 0$ является $TRUE$
// а $0 \leq j < 0$ в кванторе количества дает $FALSE$, а значит квантор возвращает 0
$N \geq 0 \Rightarrow N \geq 0 \wedge 0 = 0$
// 0 = 0 является TRUE:
$N \geq 0 \Rightarrow N \geq 0$
/*--
* α ⇒ α
* ¬α ∨ α // по закону импликации
* $TRUE$ // по закону исключения третьего
---/
$TRUE$ // по α ⇒ α

2) $(\forall_i : 1 \leq i \leq N : P \wedge B_i \Rightarrow WP(S_i, P))$

$P \wedge B_1 \Rightarrow WP(S_1, P)$
$P \wedge i < N \Rightarrow WP(IF, P)$ // возьмём консеквент импликации
$WP(IF, P) \Rightarrow domain\,(B_1 \vee B_2) \wedge (B_1 \vee B_2) \wedge (B_1 \Rightarrow WP(S_1, P)) \wedge (B_2 \Rightarrow WP(S_2, P))$
$domain((even(i) \wedge odd(a[i])) \vee \neg((even) \wedge odd(a[i]))$
$\quad \wedge ((even(i) \wedge odd(a[i])) \vee \neg((even) \wedge odd(a[i]))$
$\quad \wedge (B_1 \Rightarrow WP(S_1, P)) \wedge (B_2 \Rightarrow WP(S_2, P))$
// $((even(i) \wedge odd(a[i])) \vee \neg((even) \wedge odd(a[i])) \Rightarrow TRUE$ по
//закону исключения третьего
$domain(TRUE) \wedge TRUE \wedge (B_1 \Rightarrow WP(S_1, P)) \wedge (B_2 \Rightarrow WP(S_2, P))$
// $domain(TRUE)$ тоже является $TRUE$:
$(B_1 \Rightarrow WP(S_1, P)) \wedge (B_2 \Rightarrow WP(S_2, P))$
// распишем выражение:
$(even(i) \wedge odd(a[i])) \Rightarrow WP(x, i := x + 1, i + 1,$
$\quad 0 \leq i \leq N \wedge x = (\aleph_j : 0 \leq j < i : even(i) \wedge odd(a[j])))$
$\wedge (\neg((even) \wedge odd(a[i])) ==> WP(i := i + 1,$

$0 \le i \le N \wedge x = (\aleph_j : 0 \le j < i : even(i) \wedge odd(a[j])))$

// выполняем команду подстановки в предикат P:

$(even(i) \wedge odd(a[i])) \Rightarrow -1 \le i < N \wedge x+1 = (\aleph_j : 0 \le j \le i : even(i) \wedge odd(a[j]))$
$\wedge (\neg((even) \wedge odd(a[i])) \Rightarrow -1 \le i < N \wedge x = (\aleph_j : 0 \le j \le i : even(i) \wedge odd(a[j]))$
// из первого квантора выносим 1 элемент и изменяем область j на $0 \le j < i$
// из второго квантора выносим 0 элемент и изменяем область j на $0 \le j < i$, так как
// элемент посылки не является $(even(i) \wedge odd(a[i]))$, тем который нам нужен

$(even(i) \wedge odd(a[i])) \Rightarrow -1 \le i < N \wedge x+1 = (\aleph_j : 0 \le j < i : even(i) \wedge odd(a[j])) + 1$
$\wedge (\neg((even) \wedge odd(a[i])) \Rightarrow -1 \le i < N \wedge x = (\aleph_j : 0 \le j < i : even(i) \wedge odd(a[j])) + 0$
// сокращаем единицы в уравнении с первым квантором
// и получим, что правые части импликаций равны:

$(even(i) \wedge odd(a[i])) \Rightarrow -1 \le i < N \wedge x = (\aleph_j : 0 \le j < i : even(i) \wedge odd(a[j]))$
$\wedge (\neg((even) \wedge odd(a[i])) \Rightarrow -1 \le i < N \wedge x = (\aleph_j : 0 \le j < i : even(i) \wedge odd(a[j]))$
// обозначим левую часть через α, а правую через β и докажем
$(\alpha \Rightarrow \beta) \wedge (\neg\alpha \Rightarrow \beta)$
/*---
* $(\alpha \Rightarrow \beta) \wedge (\neg\alpha \Rightarrow \beta)$
* $(\neg\alpha \vee \beta) \wedge (\alpha \vee \beta)$ // по закону импликации
* $(\neg\alpha \vee \alpha) \vee \beta$ // дистрибутивность дизъюнкции
* //$(\neg\alpha \vee \alpha) = FALSE$ по закону противоречия
* $FALSE \vee \beta$
* β // по закону упрощения дизъюнкции
---/
// по $(\alpha \Rightarrow \beta) \wedge (\neg\alpha \Rightarrow \beta)$ останется только правая часть импликации:
$WP(IF, P) \Rightarrow -1 \le i < N \wedge x = (\aleph_j : 0 \le j < i : even(i) \wedge odd(a[j]))$
// вернемся к исходному выражению $P \wedge i < N \Rightarrow WP(IF, P)$
$0 \le i \le N \wedge x = (\aleph_j : 0 \le j < i : even(i) \wedge odd(a[j])) \wedge i < N$
$\Rightarrow -1 \le i < N \wedge x = (\aleph_j : 0 \le j < i : even(i) \wedge odd(a[j]))$
// объединим $0 \le i \le N$ и $i < N$, получим:
$0 \le i < N \wedge x = (\aleph_j : 0 \le j < i : even(i) \wedge odd(a[j]))$
$\Rightarrow -1 \le i < N \wedge x = (\aleph_j : 0 \le j < i : even(i) \wedge odd(a[j]))$
// правую часть распишем по закону дистрибутивности конъюнкции:
$0 \le i < N \wedge x = (\aleph_j : 0 \le j < i : even(i) \wedge odd(a[j]))$
$\Rightarrow 0 \le i < N \wedge x = (\aleph_j : 0 \le j < i : even(i) \wedge odd(a[j]))$
$\vee (i = -1 \wedge x = (\aleph_j : 0 \le j < i : even(i) \wedge odd(a[j]))$
// получим, что один из дизъюнктов консеквента однозначно равен антецеденту выражения
// и поэтому по НЛЛ 2 $\alpha \Rightarrow \alpha \vee \beta$ мы получаем:
$TRUE$

3) $P \wedge \neg BB \Rightarrow R$

$0 \le i \le N \wedge x = (\aleph_j : 0 \le j < i : even(i) \wedge odd(a[j])) \wedge \neg(i < N)$
$\Rightarrow x = (\aleph_j : 0 \le j < N : even(i) \wedge odd(a[j]))$

// объединим $0 \le i \le N$ и $i \ge N$, получим:

$i \ge 0 \wedge i = N \wedge x = (\aleph_j : 0 \le j < N : even(i) \wedge odd(a[j]))$

$\Rightarrow x = (\aleph_j : 0 \le j < N : even(i) \wedge odd(a[j]))$

$TRUE$ // по НЛЛ 1: $\alpha \wedge \beta \Rightarrow \alpha$

4) $P \wedge BB \Rightarrow t > 0$

$0 \le i \le N \wedge x = (\aleph_j : 0 \le j < i : even(i) \wedge odd(a[j])) \wedge (i < N) \Rightarrow N - i > 0$

// объединим $0 \le i \le N$ и $i < N$, получим:

$i < N \wedge i \ge 0 \wedge x = (\aleph_j : 0 \le j < i : even(i) \wedge odd(a[j])) \Rightarrow i < N$

// получаем, что один из конъюнктов левой части однозначно совпадает с

// консеквентом импликации, значит применяем НЛЛ 1: $\alpha \wedge \beta \Rightarrow \alpha$

$TRUE$

5) $(\forall_i : 1 \le i \le N : P \wedge B_i \Rightarrow WP("t_0 := t, S_i", t < t_0))$

$P \wedge B_1 \Rightarrow WP("t_0 := t, S_1", t < t_0)$

// распишем консеквент импликации

$WP("t_0 := N - i, IF", N - i < t_0)$

// по команде последовательности получим:

$WP(t_0 := N - i, WP(IF, N - i < t_0))$

// распишем выражение, в которое будем выполнять подстановку

$WP(IF, N - i < t_0) \Rightarrow domain((even(i) \wedge odd(a[i])) \vee \neg((even) \wedge odd(a[i])))$

$\wedge ((even(i) \wedge odd(a[i])) \vee \neg((even) \wedge odd(a[i])))$

$\wedge ((even(i) \wedge odd(a[i])) \Rightarrow WP(S_1, N - i < t_0))$

$\wedge (\neg((even) \wedge odd(a[i])) \Rightarrow WP(S_2, N - i < t_0))$

// первые 2 конъюнкта $domain(BB)$ и BB становятся $TRUE$ и мы получим:

$((even(i) \wedge odd(a[i])) \Rightarrow WP(x, i := x + 1, i + 1, N - i < t_0))$

$\wedge (\neg((even) \wedge odd(a[i])) \Rightarrow WP(i := i + 1, N - i < t_0))$

// выполняем подстановку:

$(even(i) \wedge odd(a[i])) \Rightarrow N - i - 1 < t_0$

$\wedge \neg((even) \wedge odd(a[i])) \Rightarrow N - i - 1 < t_0$

// получаем, что правые части импликаций равны

// воспользуемся ранее доказанной леммой $(\alpha \Rightarrow \beta) \wedge (\neg\alpha \Rightarrow \beta)$ и получим:

$WP(IF, N - i < t_0) \Rightarrow N - i - 1 < t_0$

// вернемся к исходному выражению

$P \wedge B_1 \Rightarrow WP(t_0 := N - i, N - i - 1 < t_0)$

// выполняем подстановку

$P \wedge B_1 \Rightarrow N - i - 1 < N - i$

// консеквент является $TRUE$:

$P \wedge B_1 \Rightarrow TRUE$

// докажем $\alpha \Rightarrow TRUE$

```
/*-----------------------------------------------------
*       α ⇒ TRUE
*       ¬α ∨ TRUE
*       // по закону упрощения дизъюнкции
*       TRUE
*-----------------------------------------------------*/
```

// окончательно получим по $\alpha \Rightarrow TRUE$
$TRUE$