

Dowodzenie poprawności programów

Wstęp do Informatyki i Programowania

Maciek Gębala

20 listopada 2025

Maciek Gębala Dowodzenie poprawności programów

Mnożenie

Chcemy pomnożyć dwie liczby całkowite a i b posługując się tylko dodawaniem, dzieleniem przez 2 i sprawdzaniem nieparzystości.

```
1:  $x \leftarrow a, y \leftarrow b, z \leftarrow 0$      $\{x = a \wedge y = b \wedge z = 0\}$ 
2: while  $x \neq 0$  do
3:    $\{z + xy = ab\}$ 
4:   if  $odd(x)$  then
5:      $z \leftarrow z + y$ 
6:   end if
7:    $\{(odd(x) \wedge z - y + xy = ab) \vee (\neg odd(x) \wedge z + xy = ab)\}$ 
8:    $x \leftarrow \lfloor x/2 \rfloor$ 
9:    $\{z + 2xy = ab\}$ 
10:   $y \leftarrow y + y$ 
11:   $\{z + xy = ab\}$ 
12: end while
13:  $\{z = ab\}$ 
```

Maciek Gębala Dowodzenie poprawności programów

Mnożenie

Niezmiennik $z + xy = ab$ inicjalizuje się dobrze po początkowych przypisaniach $x \leftarrow a, y \leftarrow b$ i $z \leftarrow 0$.

Z niezmiennika $z + xy = ab$ i zaprzeczenia warunku pętli $x = 0$ wynika natychmiast warunek końcowy $z = ab$.

Niezmiennik $z + xy = ab$ odzwierciedla się po każdym obrocie pętli. Jeśli na początku pętli x jest parzyste, to pierwsza instrukcja jest równoważna pustej i w oczywisty sposób niezmiennik pozostaje prawdziwy. Jeśli natomiast na początku pętli x jest nieparzyste, to najpierw wykona się instrukcja $z \leftarrow z + y$, po której będzie spełniony warunek $z - y + xy = ab$. Teraz wykonanie instrukcji $x \leftarrow \lfloor x/2 \rfloor$ spowoduje, że będzie spełniony warunek $z - y + (2x + 1)y = ab$, co jest równoważne warunkowi $z + 2xy = ab$, co po przypisaniu $y \leftarrow y + y$ doprowadza nas z powrotem do niezmiennika $z + xy = ab$, a to jest odtworzony niezmiennik.

Maciek Gębala Dowodzenie poprawności programów

Mnożenie

```
1:  $x \leftarrow a, y \leftarrow b, z \leftarrow 0$ 
2: while  $x \neq 0$  do
3:   if  $odd(x)$  then
4:      $z \leftarrow z + y$ 
5:   end if
6:    $x \leftarrow \lfloor x/2 \rfloor$ 
7:    $y \leftarrow y + y$ 
8: end while
```

Jest oczywiste, że jeśli $a \geq 0$ to pętla się zakończy.

Co się jednak stanie gdy $a < 0$?

Prześledźmy na tablicy działanie programu dla $a = -5$ i $b = 2$.

Program się zapętla, gdyż $\lfloor -1/2 \rfloor = -1$. Zatem program jest tylko częściowo poprawny (niezmiennik pętli $z + xy = ab$ zawsze jest prawdziwy).

Maciek Gębala Dowodzenie poprawności programów

Notatki

Notatki

Notatki

Notatki

Wyszukiwanie binarne

Założmy, że w tablicy $A(1 : n)$ mamy posortowane niemalejąco liczby całkowite i chcemy sprawdzić, czy jest w niej element x .

```
1:  $l \leftarrow 1, p \leftarrow n$ 
2: while  $l < p$  do
3:    $s \leftarrow \lfloor (l + p)/2 \rfloor$ 
4:   if  $x > A[s]$  then
5:      $l \leftarrow s + 1$ 
6:   else
7:      $p \leftarrow s$ 
8:   end if
9: end while
10: jest  $\leftarrow x = A[l]$ 
```

Nie sprawdzamy nigdzie, czy przypadkiem $A[s] = x$, tylko doprowadzamy za każdym razem do momentu, w którym badany przedział jest jednoelementowy.

Maciek Gębala Dowodzenie poprawności programów

Notatki

Wyszukiwanie binarne

Ustalmy, że predykat $\text{Sort}(A)$ jest prawdziwy jeśli tablica A jest posortowana niemalejąco.

Dopiszmy teraz do pętli niezmiennik

```
1:  $l \leftarrow 1, p \leftarrow n$ 
2: while  $l < p$  do
3:    $\{\text{Sort}(A) \wedge (1 \leq l \leq p \leq n) \wedge (\exists_{1 \leq k \leq n} A[k] = x \iff$ 
4:      $\exists_{l \leq k \leq p} A[k] = x)\}$ 
5:    $s \leftarrow \lfloor (l + p)/2 \rfloor$ 
6:   if  $x > A[s]$  then
7:      $l \leftarrow s + 1$ 
8:   else
9:      $p \leftarrow s$ 
10:   end if
11: end while
12: jest  $\leftarrow x = A[l]$ 
```

Maciek Gębala Dowodzenie poprawności programów

Notatki

Wyszukiwanie binarne

Niezmiennik

$\text{Sort}(A) \wedge (1 \leq l \leq p \leq n) \wedge (\exists_{1 \leq k \leq n} A[k] = x \iff \exists_{l \leq k \leq p} A[k] = x)$

mówi nam, że

- 1 tablica jest posortowana cały czas,
- 2 indeksy l oraz p są w zakresie indeksów tablicy i l nie przekracza p
- 3 jeśli element x jest gdzieś w tablicy, to jest między indeksami l oraz p .

Zauważmy, że niezmiennik inicjalizuje się dobrze, jeśli tylko $1 \leq n$ (czyli tablica jest niepusta).

Maciek Gębala Dowodzenie poprawności programów

Notatki

Wyszukiwanie binarne

Ze względu na to, że dla $k > 1$ zachodzi $1 \leq \lfloor k/2 \rfloor < k$ otrzymujemy nierówność $l \leq s < p$ dopóki $1 \leq l < p$ i ta ostra nierówność jest bardzo istotna. Zauważmy więc, że przypisując zmiennej l wartość $s + 1$ otrzymujemy wartość ostro większą od l , a przypisując zmiennej p wartość s otrzymujemy wartość mniejszą ostro od p . Zatem $p - l$ ma wartości w zbiorze liczb naturalnych i maleje z każdym obrotem pętli. Czyli pętla zawsze się zakończy.

Jeśli $x > A[s]$, to ze względu na przechodność relacji większości mamy $x > A[k]$ dla $1 \leq k < s$, więc jeśli istnieje x w tablicy, to istnieje między $s + 1$ a p .

Jeśli zaś $x \leq A[s]$, wówczas jeśli istnieje x w tablicy, to w szczególności jest w przedziale od 1 do s , zatem przypisanie $p \leftarrow s$ nadal utrzymuje w prawdziwości ten warunek.

Maciek Gębala Dowodzenie poprawności programów

Notatki

Wyszukiwanie binarne

Pozostaje pytanie czy zaprzeczenie warunku pętli w połączeniu z niezmiennikiem da nam końcowy warunek

$$A[l] = x \iff \exists_{1 \leq k \leq n} A[k] = x?$$

Zaprzeczenie warunku to $l \geq p$. W koniunkcji z niezmiennikiem daje nam $l = p$. Z niezmiennika wiemy, że warunek $\exists_{1 \leq k \leq n} A[k] = x$ jest równoważny warunkowi $\exists_{l \leq k \leq p} A[k] = x$, ale że $l = p$, to jest to równoważne warunkowi $A[l] = x$.

Tak znaleziona wartość x będzie zawsze pierwsza od lewej.
Dlaczego?

Maciek Gębala

Dowodzenie poprawności programów

Notatki

Wyszukiwanie binarne

Załóżmy, że instrukcję

```
1: if  $x > A[s]$  then
2:    $l \leftarrow s + 1$ 
3: else
4:    $p \leftarrow s$ 
5: end if
```

zastąpimy instrukcją

```
1: if  $x \geq A[s]$  then
2:    $l \leftarrow s$ 
3: else
4:    $p \leftarrow s - 1$ 
5: end if
```

Sytuacja wygląda na zupełnie symetryczną.

Jest to bardzo częsty błąd popełniany przy kodowaniu tego algorytmu. Program może się zapętlić. Kiedy i dlaczego?

Maciek Gębala

Dowodzenie poprawności programów

Notatki

Notatki

Notatki