

1 Przykład dowodu poprawności programu

Rozważamy program $\mathcal{P}$ (nieco zmodyfikowany w porównaniu z programem z zad 6 z listy 8):

- 1) **Dane:** tablica $\mathbf{a}$ indeksowana liczbami od 0 do $n - 1$ i liczba $\mathbf{w}$.
- 2) $\mathbf{l} = \mathbf{p} = 0; \mathbf{s} = \mathbf{a}[0];$
- 3) **while** ($\mathbf{s} \neq \mathbf{w}$ oraz $\mathbf{p} < \mathbf{n} - 1$) {
- 4) **if** ($\mathbf{s} < \mathbf{w}$) { $\mathbf{s} = \mathbf{s} + \mathbf{a}[\mathbf{p}]; \mathbf{p}++$ }
- 5) **else** { $\mathbf{s} = \mathbf{s} - \mathbf{a}[\mathbf{l}]; \mathbf{l}++$ }
- 6) }
- 7) **while** ($\mathbf{s} > \mathbf{w}$) {
- 8) $\mathbf{s} = \mathbf{s} - \mathbf{a}[\mathbf{l}]; \mathbf{l}++$
- 9) }

oraz jego specyfikację

$$\{(\forall i < n \ 0 \leq a[i]) \text{ oraz } 0 < w\} \mathcal{P}$$

$$\{\text{jeżeli } \exists x, y \ (x \leq y < n \wedge \sum_{i=x}^y a[i] = w), \text{ to } s = w \text{ oraz } s \neq w \text{ w przeciwnym razie.}\}$$

Pokażemy poprawność tej specyfikacji. Zrobimy to pokazując poprawność dwóch innych specyfikacji: pierwszej

$$\{\forall i < n \ 0 \leq a[i], \ 0 < w, \ 0 \leq L \leq P < n \text{ oraz } \sum_{i=L}^P a[i] = w\} \mathcal{P} \{s = w\}$$

oraz drugiej

$$\{\forall i < n \ 0 \leq a[i], \ 0 < w \text{ oraz } \neg \exists x, y \ (x \leq y < n \wedge \sum_{i=x}^y a[i] = w)\} \mathcal{P} \{s \neq w\}.$$

Mamy nadzieję, że jest przynajmniej intuicyjnie jasne, że przedstawiony plan jest poprawny. Polega on na rozważeniu dwóch przypadków: albo istnieją x i y takie, że $x \leq y < n$ oraz $\sum_{i=x}^y a[i] = w$, albo nie. Zauważmy też, że podczas wykonywania programu $\mathcal{P}$ nie ulegają zmianie zarówno tablica $\mathbf{a}$, jak i wartość zmiennej $\mathbf{w}$. Tak więc, jeżeli takie x i y istnieją przed wykonaniem $\mathcal{P}$, to istnieją także po wykonaniu. I jest analogicznie, jeżeli nie istnieją.

1.1 Poprawność pierwszej pomocniczej specyfikacji

Weźmy dwie formuły

$$\Phi \equiv l \leq L \wedge p \leq P \wedge s = \sum_{i=l}^p a[i] \wedge l \leq p + 1$$

oraz

$$\Psi \equiv l \leq L \wedge s = \sum_{i=l}^p a[i].$$

Zauważmy, że

Lemat 1.1 Z założeń $\forall i < n \ a[i] \geq 0$ oraz $\sum_{i=L}^P a[i] = w > 0$ i z formuły Φ wynikają następujące stwierdzenia:

- 1) $s > 0 \Rightarrow l \leq p$,
- 2) $s \neq w \Rightarrow l \neq L \vee p \neq P$,
- 3) $s > w \wedge p \leq P \Rightarrow l < L$,
- 4) $s < w \wedge l \leq L \Rightarrow p \neq P$.

Dowód. Lemat jest chyba oczywisty (wskazane są dowody metodą nie wprost), a drobne wyjaśnienia są potrzebne najwyżej w przypadku własności 1). Dotyczy ona sumy $\sum_{i=l}^p a[i]$, która w jednym przypadku ma postać $\sum_{i=p+1}^p a[i]$. Wtedy jest to suma pustego zbioru składników i należy przyjąć, że jest równa 0. $\square$

Lemat 1.2 Jeżeli $\forall i < n \ 0 \leq a[i]$, $P < n$ oraz $\sum_{i=L}^P a[i] = w > 0$, to formuła Φ jest niezmiennikiem pętli **while** z punktu 3) programu $\mathcal{P}$.

Dowód. Mamy więc wykazać, że przy podanych założeniach zachodzi następujące stwierdzenie:

$$\{s \neq w \wedge p < n - 1 \wedge \Phi\} \text{ if } (s < w) \{s = s + a[p]; p++\} \text{ else } \{s = s - a[l]; l++\} \{\Phi\}.$$

Łatwo zauważyć, że ta instrukcja warunkowa tak zmienia wartości zmiennych s , l i p , że jest zachowywany trzeci człon formuły Φ . Jeżeli wykonanie instrukcji zwiększa wartość p , to zachowywany jest też czwarty człon Φ . Jeżeli jednak jest zwiększana wartość l , to z lematu 1.1 p. 1) otrzymujemy silniejsze oszacowanie $l \leq p$ i po zwiększeniu l nadal będzie zachodzić $l \leq p + 1$. Fakt, że zachowywane są dwa pierwsze oszacowania wartości l i p wynika z trzech ostatnich punktów lematu 1.1. Stwierdzajż one, że w każdym z przypadków zachodzą silniejsze oszacowania, pozwalające na zwiększenie wartości zmiennych z zachowaniem dowodzonych oszacowań. $\square$

Lemat 1.3 Jeżeli $\forall i < n \ 0 \leq a[i]$, $p = P < n$ oraz $\sum_{i=L}^P a[i] = w > 0$, to formuła Ψ jest niezmiennikiem pętli **while** z punktu 7) programu $\mathcal{P}$.

Dowód. Podobny do dowodu poprzedniego lematu. $\square$

Lemat 1.4 Program $\mathcal{P}$ spełnia pierwszą, pomocniczą specyfikację.

Dowód. Załóżmy, że $0 \leq a[i]$ dla wszystkich $i < n$, $0 \leq L \leq P < n$ oraz $\sum_{i=L}^P a[i] = w > 0$. Nietrudno zauważyć, że w tej sytuacji, po wykonaniu podstawień z punktu 2) z programu $\mathcal{P}$ formuła Φ staje się prawdziwa.

Wobec tego i na mocy lematu 1.2, po wykonaniu instrukcji **while** z punktu 3) zachodzi formuła Φ , a także negacja warunku z tej pętli, a więc przynajmniej jedna z formuł $s = w$ lub $p \geq n - 1$.

Jeżeli $s = w$, to dalej nie są już wykonywane żadne podstawienia i ta równość zachodzi także po zakończeniu działania programu.

Jeżeli $n - 1 \leq p$, to także $p \leq P < n$ (gdyż zakładamy również, że Φ) i, w konsekwencji, mamy $n - 1 = p = P$. Zachodzi też formuła Ψ (gdyż jest konsekwencją formuły Φ), która na mocy lematu 1.3 jest niezmiennikiem instrukcji **while** z punktu 7. Wobec tego, po zakończeniu działania programu zachodzi formuła Ψ , negacja warunku z instrukcji **while**, czyli stwierdzenie $s \leq w$, oraz równości $n - 1 = p = P$ (gdyż instrukcja z punktu 7 nie zmienia wartości p).

Zachodzą więc nierówność $l \leq L$ oraz równości $p = P$ i $s = \sum_{i=l}^p a[i]$. Stąd mamy

$$s = \sum_{i=l}^p a[i] = \sum_{i=l}^P a[i] \geq \sum_{i=L}^P a[i] = w.$$

Ponieważ zachodzi także nierówność $s \leq w$, więc mamy $s = w$. $\square$

1.2 Poprawność drugiej pomocniczej specyfikacji

Poprawność drugiej specyfikacji wynika z dwóch spostrzeżeń. Po pierwsze, własność

$$\neg \exists x, y (x \leq y < n \wedge \sum_{i=x}^y a[i] = w)$$

jest prawdziwa zawsze podczas wykonywania programu $\mathcal{P}$, jeżeli tylko jest prawdziwa przed jego uruchomieniem. Wynika to stąd, że program $\mathcal{P}$ w żadnym momencie nie zmienia wartości zmiennej w i tablicy a .

Po drugie, równość

$$s = \sum_{i=l}^p a[i]$$

jest zachowywana przez obie pętle z programu $\mathcal{P}$ (jest niezmiennikiem tych pętli).

Powyższa równość jest prawdziwa przed wykonaniem pętli z punktu 3). Jest więc prawdziwa także po wykonaniu tej pętli, czyli przed wykonaniem pętli z punktu 7). Jako niezmiennik tej pętli jest też prawdziwa po jej wykonaniu, na końcu programu.

Obie wspomniane własności, razem implikują, że $s \neq w$. Oznacza to, że zachodzi także druga z rozważanych specyfikacji.