SEMANTYKA I WERYFIKACJA - ĆW. 15

Dowodzenie poprawności całkowitej.

1. Trójki Hoare'a w dowodzeniu poprawności częściowej piszemy $\{\phi\}S\{\psi\}$, co oznacza: *jeśli przed S zachodzi ϕ i S się zatrzymuje, to potem zachodzi ψ.*

   Dla poprawności całkowitej piszemy $[\phi]S[\psi]$, co oznacza *jeśli przed S zachodzi ϕ, to S się zatrzymuje i zachodzi ψ.*

   Wszystkie reguły Hoare'a dla takich trójek są jak poprzednio, poza regułą dla `while` :

   $$\frac{\phi(l+1) \Rightarrow b \qquad [\phi(l+1)]\ S\ [\phi(l)] \qquad \phi(0) \Rightarrow \neg b}{[\exists l \in \mathbb{N}.\phi(l)]\ \texttt{while}\ b\ \texttt{do}\ S\ [\phi(0)]}$$

   gdzie l jest zmienną, która nie występuje w b ani w S. Intuicja: l to licznik, który odmierza liczbę iteracji, która pozostała do końca programu. Formuła ϕ to niezmiennik pętli uzupełniony o wyliczenie wartości licznika l. Oprócz zupełnie trywialnych programów, zapisanie tego wyliczenia formułą logiczną jest praktycznie bardzo trudne.

   Inna wersja tej reguły, łatwiejsza do stosowania:

   $$\frac{l > 0 \wedge \phi(l) \Rightarrow b \qquad [l > 0 \wedge \phi(l)]\ S\ [\phi(k)] \qquad k < l \qquad \phi(0) \Rightarrow \neg b}{[\exists l \in \mathbb{N}.\phi(l)]\ \texttt{while}\ b\ \texttt{do}\ S\ [\phi(0)]}$$

2. **Zadanie:** Pokazać całkowitą poprawność potęgowania binarnego (program sprzed dwóch tygodni):

   $[y \geq 0]$
   ```
   z := 1; p := x; q := y;
   while q <> 0 do
     if odd(q) then
       z := z * p;
     q := q div 2;
     p := p * p
   ```
   $[z = x\hat{}y]$

   Rozwiązanie: Poprzednio niezmiennik pętli to było $z \cdot p^q = x^y$. Teraz uzupełniony "niezmiennik z licznikiem" to:

   $$\phi(l) = (z \cdot p^q = x^y) \wedge (q = l)$$

   Dowód idzie tak jak poprzednio, a dla treści pętli pokazujemy:

   $$[(z \cdot p^q = x^y) \wedge q = l \wedge l > 0]\ S\ [(z \cdot p^q = x^y) \wedge q = l\ \texttt{div}\ 2]$$

3. Inna, jeszcze wygodniejsza metoda: żeby pokazać

$$[\phi]\ \texttt{while}\ b\ \texttt{do}\ S\ [\phi \wedge \neg b],$$

pokaż

$$[\phi \wedge b]\ S\ [\phi]$$

(czyli zastosuj dokładnie tę samą regułę co dla częściowej poprawności), ale dodatkowo:

- znajdź dobrze ufundowany porządek częściowy W i funkcję ze stanów pamięci w W,
- takie że każde wykonanie ciała pętli powoduje zmniejszenie wartości tej funkcji.

Przykłady porządków dobrze ufundowanych:

- $\mathbb{N}$,
- $\mathbb{N} \times \mathbb{N}$ uporządkowane po współrzędnych,
- $\mathbb{N} \times \mathbb{N}$ uporządkowane leksykograficznie.

$\mathbb{N}^*$ uporządkowane leksykograficznie nie jest dobrze ufundowane.

Program, w którym można zastosować porządek leksykograficzny:

```
while x>0 do
  if y>0 then
    y := y-1
  else
    x := x-1;
    y := f(x)
```

Ten program się zatrzymuje niezależnie od tego, jak działa (zatrzymująca się) funkcja f.

4. **Zadanie:** Zrobić ten sam program co poprzednio, tą nową metodą.

   Rozwiązanie: Niezmiennik ϕ jest jak dwa tygodnie temu, malejące wyrażenie opisujemy następującą notacją:

```
[y ≥ 0]
z := 1; p := x; q := y;
while [φ] q <> 0 do [decr q in ℕ wrt ≤]
  if odd(q) then
    z := z * p;
  q := q div 2;
  p := p * p
[z = x^y]
```

Innymi słowy, normalnie pokazujemy poprawność częściową, a dla poprawności całkowitej organiczamy się do wskazania malejącego wyrażenia i porządku, w którym ono maleje.

5. **Zadanie:** Pokaż własność stopu programu:

```
{n ≥ 0}
s := 1; i := 1;
while i < n do
  k := 1; m := 0; p := 1;
  while p < i do
    k := k + 1;
    if m < p then
      m := m + 1;
      p := 1
    else
      p := p + 1;
  s := s + 6 * k + p - l;
  i := i + 1;
{s = n^3}
```

Własność stopu pętli zewnętrznej łatwo jest pokazać, gdyż w każdym obiegu rośnie wartość zmiennej i, czyli maleje wartość wyrażenia $n - i$.

Dodatkowo $n - i \geq 0$ trzeba zagwarantować pokazując niezmiennik.

Trudniej jest pokazać własność stopu pętli wewnętrznej.

Choć dozór tej pętli to $p < i$, to zasadnicze znaczenie dla tej pętli ma nie tyle wzrost wartości zmiennej p (zmienna i nie zmienia się w tej pętli) co wzrost wartości zmiennej m. Ale wyrażenie $i - m$ nie maleje po każdym obiegu pętli! Może ono pozostać niezmienione, ale wtedy rośnie na pewno wartość zmiennej p, czyli maleje $i - p$. Widać zatem, że warto tutaj wziąć pod uwagę porządek leksykograficzny par liczb: para wartości wyrażeń $(i - p, i - m)$ maleje w porządku leksykograficznym po każdym obiegu pętli. Obydwa mają przy tym zawsze wartość większą lub równą 0, co trzeba pokazać niezmiennikiem.

A jak pokazać poprawność częściową?

Wskazówki: $(n + 1)^3 = n^3 + 3n^2 + 3n + 1$, a $1 + 2 + \cdots + m = \frac{m(m+1)}{2}$.

Niezmiennik pętli zewnętrznej: $s = i^3 \wedge 1 \leq i \leq n$.

Niezmiennik pętli wewnętrznej:

$$s = i^3 \wedge i < n \wedge k = (1 + 2 + \cdots + l) + p \wedge 1 \leq p \leq l + 1 \leq i$$

6. To teraz taki program. Precondition: pierwsza połowa tablicy jest posortowana i druga też. Postcondition: wszystko w pierwszej połowie jest większe od wszystkiego w drugiej.

```
i := 1; j := 2 * n;
while A[i] < A[j] do
  "zamień A[i] z A[j]";
  i := i + 1;
  j := j - 1;
```

Rozwiązanie: malejąca miara to $j - i$, niezmiennik łatwo wymyślić.