SEMANTYKA I WERYFIKACJA - ĆW. 13

Zaczynamy weryfikację programów.

1. Logika Hoare'a:

$$\frac{}{\{\phi\}\ \texttt{skip}\ \{\phi\}} \qquad \frac{}{\{\phi[x \mapsto e]\}\ x := e\ \{\phi\}} \qquad \frac{\{\phi\}S_1\{\theta\} \quad \{\theta\}S_2\{\psi\}}{\{\phi\}S_1; S_2\{\psi\}}$$

$$\frac{\{\phi \wedge b\}\ S_1\ \{\psi\} \quad \{\phi \wedge \neg b\}\ S_2\ \{\psi\}}{\{\phi\}\ \texttt{if}\ b\ \texttt{then}\ S_1\ \texttt{else}\ S_2\ \{\psi\}} \qquad \frac{\{\phi \wedge b\}\ S_1\ \{\phi\}}{\{\phi\}\ \texttt{while}\ b\ \texttt{do}\ S\ \{\phi \wedge \neg b\}}$$

$$\frac{\phi' \Rightarrow \phi \quad \{\phi\}\ S\ \{\psi\} \quad \psi \Rightarrow \psi'}{\{\phi'\}\ S\ \{\psi'\}}$$

2. **Zadanie**: Przeprowadź dowód poprawności częściowej programu:

$\{n \geq 1\}$
```
x := 1; y := 0;
while x <= n do
  y := y + x;
  x := x + 1
```
$\{y = \frac{n(n+1)}{2}\}$

Rozwiązanie:

- Na początek wstwmy asercję po pierwszych dwóch przypisaniach, "przeciągając" asercję początkową przez te przypisania:

  $\{n \geq 1\}$
  ```
  x := 1; y := 0;
  ```
  $\{n \geq 1 \wedge x = 1 \wedge y = 0\}$
  ```
  while x <= n do
    y := y + x;
    x := x + 1
  ```
  $\{y = \frac{n(n+1)}{2}\}$

- Proste ćwiczenie: pokazać, że dwa początkowe przypisania pokazują powstałą trójkę.
- Teraz trzeba wymyślić niezmiennik pętli N taki, że:
  - $n \geq 1 \wedge x = 1 \wedge y = 0 \Rightarrow N$
  - $N \wedge x > n \Rightarrow y = \frac{n(n+1)}{2}$
  - $\{N\}$
    ```
    while x <= n do
      y := y + x;
      x := x + 1
    ```
    $\{N \wedge x > n\}$

Do tej ostatniej trójki potrzebujemy:

$\{N \ \wedge \ x \leq n\}$
`y := y + x;`
`x := x + 1`
$\{N\}$

- Zgadujemy niezmiennik:

$$N \quad \equiv \quad y = \frac{x(x-1)}{2} \ \wedge \ x \leq n+1$$

- Sprawdzamy, że wszystko się zgadza.

Pełne rozwiązanie (konwencja: niezmiennik pętli wpisujemy po `while`):

$\{n \ \geq \ 1\}$
`x := 1;`
$\{n \ \geq \ 1 \ \wedge \ x = 1\}$
`y := 0;`
$\{n \ \geq \ 1 \ \wedge \ x = 1 \ \wedge \ y = 0\}$
$\Downarrow$
$\{N\}$
`while` $\{N\}$ `x <= n do`
$\{N \ \wedge \ x \leq n\}$
`y := y + x;`
$\{y \ = \ \frac{x(x+1)}{2} \ \wedge \ x \ \leq \ n \ \}$
`x := x + 1`
$\{N \ \wedge \ x \leq n+1\} \ = \ \{N\}$
$\{y \ = \ \frac{n(n+1)}{2}\}$

A oto program początkowy i po uzupełnieniu przez asercje w Haha:

```
function sum ( n : Z ) : Z
    var x : Z
        y : Z
begin
    x := 1
    y := 0
    while x <= n do
    begin
      y := y + x
      x := x + 1
    end
    sum := y
end
```

```
function sum ( n : Z ) : Z
    precondition natural: n >= 1
    postcondition gauss: sum = n * (n+1) / 2
    var x : Z
        y : Z
begin
    x := 1
    { n  >= 1 /\ x = 1 }
    y := 0
    { n >= 1 /\ x = 1 /\ y = 0 }
    while x <= n do
    invariant y = x * (x-1)/2 /\ x <= n+1
    begin
      y := y + x
      { y = x * (x+1)/2 /\ x<=n}
      x := x + 1
    end
    { y = n * (n+1) / 2 }
    sum := y
end
```

3. **Zadanie:** Uzupełnij konkluzje reguł przez wymyślenie ich przesłanek:

$$\frac{\alpha \Rightarrow \beta[x \mapsto e]}{\{\alpha\}\ x := e\ \{\beta\}} \qquad \frac{\alpha \Rightarrow \gamma \quad \{\gamma \wedge b\}\ S\ \{\gamma\} \quad \gamma \wedge \neg b \Rightarrow \beta}{\{\alpha\}\texttt{while}\ b\ \texttt{do}\ S\ \{\beta\}}$$

4. **Zadanie:** Dowiedź poprawności częściowej programu wykonującego potęgowanie binarne:

```
{y ≥ 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^y}
```

gdzie `odd` sprawdza czy liczba jest nieparzysta.

Rozwiązanie: Niezmiennik to

$$z \cdot p^q = x^y$$

5. Zaproponuj regułę dla instrukcji `repeat` I `until` b.

Rozwiązanie: Można na przykład:

$$\frac{\{\alpha\}\ I\ \{\beta\} \quad \{\beta \wedge \neg b\}\ I\ \{\beta\}}{\{\alpha\}\ \texttt{repeat}\ I\ \texttt{until}\ b\{\beta \wedge b\}}$$

Ta reguła odwołuje się do I dwa razy. Prostsza reguła:

$$\frac{\{\beta\}\ I\ \{\beta\}}{\{\beta\}\ \texttt{repeat}\ I\ \texttt{until}\ b\{\beta \wedge b\}}$$

jest słabsza, bo we wnętrzu pętli nie można w niej korzystać z tego, że począwszy od drugiej iteracji nie zachodzi b. Jeszcze inna możliwość:

$$\frac{\{\alpha\}\ I\ \{\beta\} \quad \beta \wedge \neg b \Rightarrow \alpha}{\{\alpha\}\ \texttt{repeat}\ I\ \texttt{until}\ b\{\beta \wedge b\}}$$

Ćwiczenie: wyprowadź tę ostatnią regułę z tej pierwszej.

6. Napisz program obliczający największy wspólny dzielnik dwóch liczb i przeprowadź dowów poprawności częściowej.