Reguły wnioskowania dla poprawności częściowej

Logika Hoare'a

$\{\text{true}\}$
$x := 1;\ n := N;$
WHILE $n > 0$ DO $\quad \{x \cdot n! = N!\}$
BEGIN
$x := x \cdot n;\ n := n - 1$
END
$\{x = N!\}$

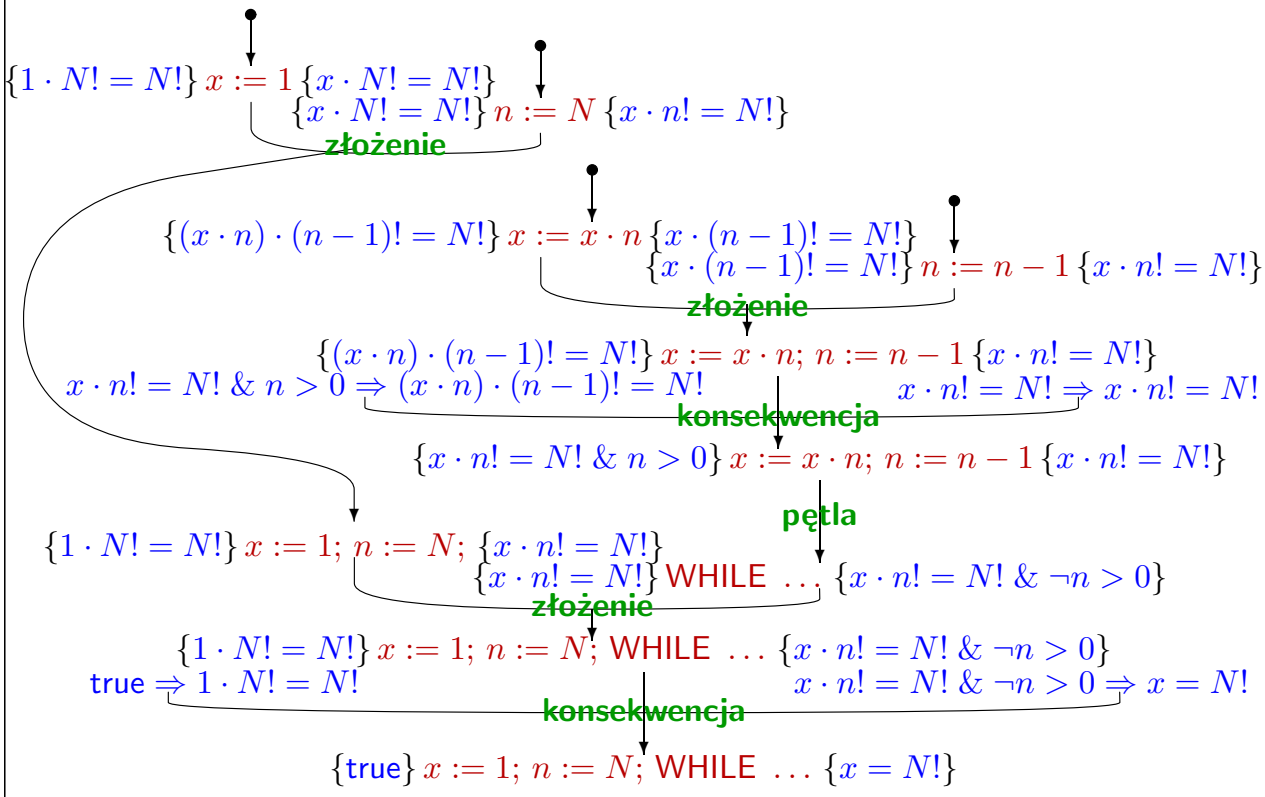
niezmiennik pętli

Reguły wnioskowania dla poprawności częściowej

$\{\text{true}\}$
$x := 1;\ n := N;$
WHILE $n > 0$ DO $\quad \{x \cdot n! = N!\}$
BEGIN
$x := x \cdot n;\ n := n - 1$
END
$\{x = N!\}$

1. Zgadnąć kandydata na niezmiennik: $x \cdot n! = N!$
2. Dowieść, że to niezmiennik:
   $x \cdot n! = N!\ \&\ n > 0 \stackrel{?}{\Rightarrow} (x \cdot n) \cdot (n-1)! = N!$
3. Dowieść, że inicjalizacja gwarantuje jego spełnienie:
   $\text{true} \stackrel{?}{\Rightarrow} 1 \cdot N! = N!$
4. Dowieść, że niezmiennik gwarantuje spełnienie warunku końcowego:
   $x \cdot n! = N!\ \&\ \neg\, n > 0 \stackrel{?}{\Rightarrow} x = N!$

Reguły wnioskowania dla poprawności częściowej



Reguły wnioskowania dla poprawności częściowej

- Logika Hoare'a dla częściowej poprawności jest
  - *poprawna* — wyprowadza wyłącznie prawdziwe fakty, oraz
  - *zupełna* — każde prawdziwe stwierdzenie o częściowej poprawności daje się w niej wyprowadzić (uwaga: to jest niezbyt ścisłe).
- Nieautomatyzowalne kroki w dowodach częściowej poprawności:
  - zgadywanie niezmienników pętli,
  - udowadnianie faktów nieinformatycznych potrzebnych regule konsekwencji.
- Logika Hoare'a daje się (z pewnym wysiłkiem) uogólnić na bardziej złożone języki programowania.
- Dowodzenie częściowej poprawności jest zbyt żmudne jak na codzienną praktykę programistyczną.
- Logika Hoare'a wywarła wpływ na rozwój języków programowania oraz metod programowania.

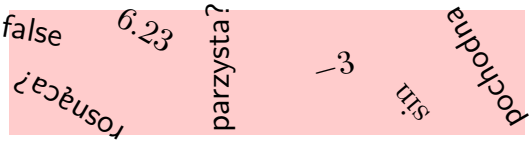
Wnioskowanie o typach

- Jasio ma arbuzy a Hania ma poziomki. Jasio ma ich dwa a Hania pięć. Które z nich ma więcej owoców?
- Powierzchnia Ziemi wynosi 510 mln km^2 a jej odległość od Słońca wynosi 144 mln km. Co jest większe?
- Mam trzy kilo wolnego czasu, ale ani sekundy pieniędzy.
- Jeśli za zmienną całkowitą n podstaw dwukrotną literę `'a'`, to powiększ napis `"PWSZ"` o jeden i wydrukuj.

```
main()
{
 int n;
 if (n = 2 * 'a')  printf("%s\n", "PWSZ"+1);
}
```

KONFLIKTY TYPÓW!

Wnioskowanie o typach



JĘZYK BEZTYPOWY

.....

rosnąca? : $(\mathbb{R} \to \mathbb{R}) \to \mathbb{B}$ $\qquad$ $\frac{d}{dx} : (\mathbb{R} \to \mathbb{R}) \to (\mathbb{R} \to \mathbb{R})$

$\sin : \mathbb{R} \to \mathbb{R}$ $\qquad$ parzysta? : $\mathbb{I} \to \mathbb{B}$

$-3 : \mathbb{I}$ $\qquad$ $6.23 : \mathbb{R}$ $\qquad$ *false* : $\mathbb{B}$

JĘZYK UTYPOWANY

$\mathbb{I}$ — całkowite $\qquad$ $\mathbb{R}$ — rzeczywiste $\qquad$ $\mathbb{B}$ — logiczne

Reguły wnioskowania dla systemu typów

$\{x_1 : T_1, \ldots, x_k : T_k\} \vdash t : T$ — jeśli zmienne $x_1, \ldots, x_k$ są typów (odpowiednio) $T_1, \ldots, T_k$, to wyrażenie t *jest typu* T.

Przykład:

$\{\} \vdash$ false : boolean
$\{\} \vdash -6.35$: real
$\{x$: integer$\} \vdash x/2$: integer
$\{x$: real$\} \vdash x/2$: real
$\{x$: real$\} \vdash y/2$: — BŁĄD
$\{x$: integer, y : boolean$\} \vdash x + y$: — BŁĄD
$\{x$: integer, y : integer$\} \vdash$ (IF $x < y$ THEN x ELSE y) : integer
$\{x$: integer, y : integer$\} \vdash$ (IF $x < y$ THEN 1 ELSE false) : — BŁĄD
$\{\} \vdash (\lambda x \,.\, 2 \cdot x)$: real $\to$ real — podwojenie danej liczby

$\lambda x \,.\, t$ — funkcja, która dowolnemu x przypisuje t

$\{\} \vdash (\lambda n \,.\, \lambda x \,.\, \underbrace{x \cdot \ldots \cdot x}_{n})$: integer $\to$ (real $\to$ real) — potęga

$\{\} \vdash (\lambda f \,.\, f(3.14))$: (real $\to$ real) $\to$ real — wart. funkcji w punkcie

Reguły wnioskowania dla systemu typów

Typowanie stałych:

$$\frac{}{\{\ldots\} \vdash \text{false} : \text{boolean}} \qquad \frac{}{\{\ldots\} \vdash 4 : \text{integer}} \qquad \frac{}{\{\ldots\} \vdash 6.3 : \text{real}}$$

$$\frac{}{\{\ldots\} \vdash + : \text{real} \to \text{real} \to \text{real}} \qquad \frac{}{\{\ldots\} \vdash > : \text{real} \to \text{real} \to \text{boolean}}$$

$$\frac{}{\{\ldots\} \vdash \text{IF} : \text{boolean} \to \text{real} \to \text{real} \to \text{real}}$$

Typowanie zmiennych:

$$\frac{}{\{\ldots, x : T, \ldots\} \vdash x : T}$$

Typowanie funkcji:

$$\frac{\{\ldots, x : T_1\} \vdash t : T_2}{\{\ldots\} \vdash (\lambda x.t) : T_1 \to T_2}$$

Typowanie zastosowania funkcji do argumentu:

$$\frac{\{\ldots\} \vdash f : T_1 \to T_2 \quad \{\ldots\} \vdash t : T_1}{\{\ldots\} \vdash f(t) : T_2}$$