Pavol Jozef Šafárik University in Košice
Faculty of Science

Verifikácia programov
Hoareova metóda, pokračovanie

Gabriela Andrejková

C. A. R. Hoare

použil ako východisko na dokazovanie správnosti programov zapísaných v ľubovoľnom programovacom jazyku typu algol60, pascal, a pod., metódu **induktívnych podmienok**.
Vypracoval axiomatický prístup k definícii sémantiky príkazov.

Vychádzal z toho, že k tomu, aby proces dokazovania správnosti programu mohol byť sformalizovaný, je potrebné mať vhodný systém, v ktorom budú vyjadrované a odvodzované vlastnosti príkazov, resp. programov.

Induktívny výraz

Hoareov axiomatický prístup je najčastejšie používaným spôsobom na vyjadrenie sémantiky príkazov.

- Axiómy tohto systému definujú sémantiku najjednoduchších príkazov jazyka a
- odvodzovacie pravidlá umožňujú vyjadriť sémantiku vhodne vytvorenej postupnosti príkazov pomocou sémantiky jednoduchých príkazov, ktoré túto postupnosť vytvárajú.

Induktívny výraz

Na vyjadrenie sémantiky Hoare zaviedol tzv. **induktívne výrazy** (tiež ich budeme nazývať špecifikačné výrazy), ktoré sú tvaru

$$\{P\}\,\Pi\,\{Q\} \tag{1}$$

kde P, Q sú formuly predikátového počtu 1. rádu s rovnosťou a Π je jednoduchý príkaz alebo vhodne vytvorená postupnosť príkazov vo zvolenom programovacom jazyku.

Induktívny výraz

Tento induktívny výraz má nasledujúci význam pri dokazovaní čiastočnej správnosti programu (vychádzame z toho, že výpočet podľa príkazu alebo postupnosti príkazov skončí):

Vždy, keď formula P je splnená bezprostredne pred vykonaním Π a Π končí, potom formula Q je splnená po vykonaní Π. (2.2)

Induktívny výraz

Dokázať, že program Π je čiastočne správny vzhľadom na vstupný predikát P a výstupný predikát Q teda znamená odvodiť induktívny výraz

$$\{P\}\,\Pi\,\{Q\}.$$

Odvodzovacie pravidlá

$$\frac{P_1, P_2, \ldots, P_n}{P}, n \geq 1,$$

kde $P_1, P_2, \ldots, P_n, P$ sú tvrdenia.

Interpretácia: **"ak platia tvrdenia $P_1, P_2, \ldots, P_n$, potom platí tvrdenie P"**.
$P_1, P_2, \ldots, P_n$ sú **predpoklady**, P je **dôsledok**.

P_i, $1 \leq i \leq n$, je induktívny výraz alebo axióma priradenia alebo logický výraz, ktorého platnosť je ukázaná vopred alebo v priebehu vytvárania dôkazu.
Ak v priebehu odvodzovania sa ukázala platnosť $P_1, P_2, \ldots, P_n$, potom P je možné považovať za ďalší člen dôkazu.

Axiómy

A1. Pre prázdny príkaz

$$\frac{R \to S}{\{R\}\{S\}}$$

A2. Pre priraďovací príkaz

$$\{R[e\sharp y]\}\, y := e\, \{R\},$$

ktorý vyjadruje toto: **ak R je pravdivá predtým, než výraz e bol nahradený premennou y, potom R je tiež pravdivá, keď do y bola priradená nová hodnota, určená výrazom e.**

Výraz $R[e\sharp y]$ znamená, že každý voľný výskyt premennej y v R je nahradený výrazom e.

Totálna správnosť programov

Pretože vyššie uvedené odvodzovacie pravidlá boli vytvárané na základe platnosti (2.2) t.j. na základe predpokladu, že program končí, umožňujú tým dokazovať len čiastočnú správnosť programov.

Ak by sme význam induktívneho výrazu so špecifikáciami zmenili tak, že v každom príkaze (najdôležitejšie sú príkazy cyklu) budeme dokazovať okrem čiastočnej správnosti aj to, že program končí, a tým by sme mohli dokázať totálnu správnosť programu.

Tu induktívny výraz

$$\{P\}\,\Pi\,\{Q\}$$

bude vyjadrovať nasledovné:
Ak P platí bezprostredne pred vykonaním Π, potom výpočet podľa Π skončí a po skončení platí Q.

Pretože sa mení význam induktívneho výrazu, je potrebné urobiť zmeny aj v odvodzovacích pravidlách.

Axiómy, pravidlá pre podmienkové príkazy a pravidlá konsekvencie zostávajú v platnosti tak, ako boli uvedené, pretože priraďovací príkaz a podmienkový príkaz vždy skončia.
Je však potrebné zaviesť **nové odvodzovacie pravidlá pre príkazy cyklov**.

Pri vyjadrovaní podmienok v príkazoch sme doposiaľ vynechávali zápis premenných, pomocou ktorých boli vytvorené, pretože nebolo potrebné zdôrazniť ich.

Teraz tieto premenné zapíšeme pomocou vektora $\bar{y}$, pretože v predpokladoch odvodzovacích pravidiel budeme potrebovať jeho dve hodnoty vypočítané po sebe vykonaním príkazov cyklu.

Nové odvodzovacie pravidlá pre príkazy cyklu:

1.
$$\frac{\{P(\overline{y}) \wedge B(\overline{y})\}\,\Pi\,\{P(\overline{y}') \wedge F(\overline{y}') \sqsubset F(\overline{y})\}}{\{P(\overline{y})\}\ \text{while}\ B(\overline{y})\ \text{do}\ \Pi\ \text{od}\ \{P(\overline{y}') \wedge \neg B(\overline{y}')\}}$$

2.
$$\frac{\{P(\overline{y})\}\,\Pi\,\{Q(\overline{y}') \wedge F(\overline{y}') \sqsubset F(\overline{y})\},\ Q(\overline{y}') \wedge \neg B(\overline{y}') \rightarrow P(\overline{y}')}{\{P(\overline{y})\}\ \text{repeat}\ \Pi\ \text{until}\ B(\overline{y}')\,\{Q(\overline{y}') \wedge B(\overline{y}')\}}$$

kde F je vhodne zvolená parciálna funkcia, $F : D\overline{y} \rightarrow W$ a $(W, \sqsubset)$ je fundovaná množina, t. j. neexistuje v nej nekonečná klesajúca postupnosť.

Pomocou $\overline{y}, \overline{y}'$ sme vyjadrili starú a novú hodnotu premenných, s ktorými program Π pracuje, čo je potrebné kvôli porovnaniu hodnôt funkcie F.

V pravidlách je zahrnutá myšlienka, z ktorej vychádza aj Floydova metóda pre dokazovanie, a síce využitie funkcie, ktorej funkčné hodnoty sú vo fundovanej množine. Teda, ak ukážeme platnosť predpokladov, potom uvedený príkaz cyklu končí, čo vyplýva z vlastností funkcie F.

Použitie uvedených pravidiel uvedieme v nasledujúcom príklade, v ktorom ukážeme, že vytvorený program je totálne správny.

Príklad

Nech $A = (a_1, a_2, \ldots, a_n)$ je permutácia množiny čísel $\{1, 2, \ldots, n\}$, $n \geq 1$. Vektorom inverzií permutácie A nazývame vektor $B = (b_1, b_2, \ldots, b_n)$, kde b_j je rovné počtu prvkov väčších než j a nachádzajúcich sa v permutácii A vľavo od j.

Napríklad, permutácia $A = (6, 1, 4, 5, 2, 3)$ má vektor inverzií $B = (1, 3, 3, 1, 1, 0)$.

Nasledujúci program v jazyku JO k danému vektoru inverzií vytvára permutáciu A.

Totálna správnosť programov

```
Program Permut;
        y1 := 1;
        while   y1 ≤ n do
c1              a[y1] := 0;
                y1 := y1 + 1;
        od;
        y1 := 0;
        while   y1 < n do
                (y1, y2, y3) := (y1 + 1, 0, 0);
                while y3 < b[y1] do
c4              y2 := y2 + 1;
        c2      if a[y2] = 0 then y3 := y3 + 1;
                od;
                y2 := y2 + 1;
        c3      while a[y2] ≠ 0 do y2 := y2 + 1 od;
                a[y2] := y1;
        od;
```

Dokážeme totálnu správnosť tohto programu vzhľadom **na vstupnú podmienku**

$P(b) : b = (b_1, b_2, \ldots, b_n), n \geq 1$, je vektor inverzií permutácie s n prvkami, t. j. platí $b_{n-j} \leq j$, pre $0 \leq j \leq n-1$;

a **výstupnú podmienku**

$Q(b, a)$: a je permutácia s vektorom inverzií b, t. j.
$(\forall j)(1 \leq j \leq n)[b_j = |\{a_l : a_l > j \wedge 1 \leq l \leq k \wedge j = a_k\}|]$

Ak vo výraze $b_j = |\{a_l : a_l > j \wedge 1 \leq l \leq k \wedge j = a_k\}|$ všetky prvky väčšie ako j nahradíme hodnotou 0, dostaneme $b_j = |\{a_l : a_l = 0 \wedge 1 \leq l \leq k \wedge j = a_k\}|$. Tento posledný výraz budeme používať pri dôkaze správnosti.

Program pracuje na nasledujúcom princípe:
Permutácia má mať n prvkov, $n \geq 1, a[1], a[2], \ldots, a[n]$. Indexy budeme v ďalšom vyjadrovať v zátvorkách v súlade s jazykom JO. Neobsadené pozície, t. j. pozície, kam ešte nebol uložený žiadny prvok permutácie, označíme 0. Na začiatku teda platí

$$a[1] = a[2] = \cdots = a[n] = 0.$$

Permutáciu budeme konštruovať postupne od určenia pozície pre prvok 1, pre prvok 2, atď. $b[y1]$, pre $1 \leq y1 \leq n$, určuje, koľko väčších prvkov má byť uložených pred prvkom $y1$, a teda určuje koľko neobsadených pozícií je potrebné vynechať. Potom je potrebné zabezpečiť, aby prvok $y1$ bol uložený na nasledujúcu voľnú pozíciu.

Dôkaz totálnej správnosti je urobíme celkom formálne, aj keď aj menej skúsenému programátorovi je zrejmé, že napríklad cyklus $c1$ skončí a po jeho vykonaní platí

$$(\forall y_1)(1 \leq y_1 \leq n)[a[y_1] = 0] \wedge y_1 = n + 1.$$

Predikát

$$(\forall i)(1 \leq i < 1)a[i] = 0 \tag{2}$$

je zrejme splnený, pretože i spĺňujúce uvedené nerovnosti neexistuje.

Označme

$$R(y_1, a) : (\forall i)(1 \leq i < y_1)[a[i] = 0].$$

Teda $R(1, a)$ je vlastne (2) a platí $R(1, a) \equiv true$.

Zrejme platí

$$R(y_1, a) \wedge a[y_1] = 0 \rightarrow R(y_1 + 1, a), \quad P(b) \rightarrow R(1, a). \tag{3}$$

Podľa axiómy pre priraďovací príkaz a pravidla konsekvencie dostávame

$$\{R([1 \sharp y_1], a)\}\, y_1 := 1\, \{R(y_1, a)\}, \tag{4}$$

$$\{true\}\, y_1 := 1\, \{R(y_1, a)\}. \tag{5}$$

Na základe axiómy pre priraďovací príkaz a platnosti (2) dostávame

$$\{R(y_1, [a \square 0 \sharp a[y_1]])\}\, a[y_1] := 0\, \{R(y_1 + 1, a)\} \tag{6}$$

$$\{R([y_1 + 1 \sharp y_1], a)\}\, y_1 := y_1 + 1\, \{R(y_1, a)\} \tag{7}$$

Podľa pravidla pre kompozíciu z platnosti (6) a (7) dostávame

$$\{R(y_1, a)\} a[y_1] := 0; y_1 := y_1 + 1 \{R(y_1, a)\} \tag{8}$$

Je potrebné dokázať, že cyklus $c1$ končí, preto je potrebné zvoliť fundovanú množinu a funkciu, ktorá je klesajúca a nadobúda funkčné hodnoty zo zvolenej fundovanej množiny. Za fundovanú množinu zvolíme $(N, <)$ a funkcia $F1 : N \to N$ bude definovaná nasledovne:

$$F1(y_1) = n - y_1. \tag{9}$$

Platí $F1(y_1) > F1(y_1 + 1)$ a $F1(y_1) \geq 0$, pre $y1 \leq n$.
Aplikáciou pravidla pre cykly na (7) dostávame

$$\frac{\{R(y_1, a) \wedge y_1 \leq n\} a[y_1] := 0; y_1 := y_1 + 1 \{R(y_1, a) \wedge F1(y_1 + 1) < F1(y_1)\}}{\{R(y_1, a)\} \text{ while } y_1 \leq n \text{ do } a[y_1] := 0; y_1 := y_1 + 1 \text{ od } \{R(y_1, a) \wedge y_1 > n\}} \quad (10)$$

Použitím pravidla pre kompozíciu na (5) a (10) dostávame

$$\begin{array}{l} \{true\}\ y_1 := 1; \\ \quad \text{while} \quad y_1 \leq n \text{ do} \\ \qquad\qquad a[y_1] := 0; \\ \qquad\qquad y_1 := y_1 + 1; \\ \qquad\qquad \text{od } \{R(y_1, a) \wedge y_1 > n\} \end{array} \quad (11)$$

V dôkaze v ďalšej časti budeme používať predikát

$$S(y_1, y_2, y_3, a) : y_3 = |\{p : a[p] = 0 \wedge 1 \leq p \leq y_2\}| \wedge y_2 \leq y_3 + y_1 - 1 \tag{12}$$

Zrejme platí

$$S(y_1, 0, 0, a) \equiv true \text{ pre } 1 \leq y_1 \leq n. \tag{13}$$

Podľa axiómy pre priraďovací príkaz platí

$$\{S(y_1, [y_2 + 1 \sharp y_2], y_3, a)\}\, y_2 := y_2 + 1\, \{S(y_1, y_2, y_3, a)\} \tag{14}$$

Z vlastností predikátu S dostávame

$$S(y_1, y_2, y_3, a) \wedge a[y_2 + 1] \neq 0 \rightarrow S(y_1, y_2 + 1, y_3, a) \tag{15}$$

$$S(y_1, y_2, y_3, a) \wedge a[y_2 + 1] = 0 \rightarrow S(y_1, y_2 + 1, y_3 + 1, a) \quad (16)$$

Podľa axiómy pre príkaz priradenia dostávame

$$\{S(y_1, y_2{+}1, [y_3{+}1 \sharp y_3], a)\}\, y_3 := y_3{+}1\, \{S(y_1, y_2{+}1, y_3, a)\} \quad (17)$$

Podľa pravidla pre podmienkový príkaz a pomocou (16) a (17) dostávame

$\{S(y_1, y_2 + 1, y_3, a)\}$

$$y_2 := y_2 + 1;\ \text{ if } a[y_2] = 0 \text{ then } y_3 := y_3 + 1 fi; \quad (18)$$

$\{S(y_1, y_2, y_3, a)\}$

Za fundovanú množinu zvolíme $(N, <)$. Zvolíme funkciu $F2 : N \times N \to N$ takto:

$$F2(y_2, y_3) = n + b[y_1] - y_2 - y_3$$

pričom $b[y_1]$ sa v tomto cykle nemení. Nie je možné zvoliť $F2 : N \to N$ takto: $F2(y_3) = n - y_3$, $F2$ by bola nerastúca. Je potrebné ukázať, že $F2(y_2, y_3)$ je klesajúca a platí $F2(y_2, y_3) \geq 0$. $F2$ je klesajúca, pretože hodnota y_2 pri každom prechode rastie a hodnota y_3 neklesá.
$F2(y_2, y_3) \geq 0$ z nasledujúcich dôvodov:
Počet neobsadených pozícií je doposiaľ $y_1 - 1 \geq 0$, počet preskúmaných je y_3, preto

$$y_2 \leq y_3 + y_1 - 1 < b[y_1] + y_1 - 1, \text{ pre } y_3 < b[y_1].$$

Teda pre $y_1 \leq n$

$$F2(y_2, y_3) = n+b[y_1]-y_2-y_3 > n+b[y_1]-(b[y_1]+y_1-1) = n-y_1+1,$$

Aplikáciou pravidla pre príkaz cyklu na (18) dostávame
$\{S(y_1, y_2, y_3, a) \wedge y_3 < b[y_1]\}$
$y_2 := y_2 + 1;$
if $a[y_2] = 0$ then $y_3 := y_3 + 1;$
$\{S(y_1, y_2, y_3, a) \wedge F2(y_2 + 1, y_3') < F2(y_2, y_3)\}$

$\{S(y_1, y_2, y_3, a)\}$ while $y_3 < b[y_1]$ do $y_2 := y_2 + 1;$
if $a[y_2] = 0$ then $y_3 := y_3 + 1;$ od

$$\{S(y_1, y_2, y_3, a) \wedge y_3 = b[y_1]\} \qquad (19)$$

Teda po skončení cyklu c2 platí

$$y_2 \leq y_3+y_1-1 = b[y_1]+y_1-1 \leq n-y_1+y_1-1 = n-1, \text{ t. j. } y_2 \leq n-1.$$

Z čoho vyplýva, že medzi $a[y_2+1], \ldots, a[n]$ existuje aspoň jedna neobsadená pozícia.

$$S(y_1, y_2, y_3, a) \wedge y_3 = b[y_1] \rightarrow b[y_1] = |\{p : a[p] = 0 \wedge 1 \leq p \leq y_2| \wedge y_2 \leq n-1 \tag{20}$$

$$b[y_1] = |\{p : a[p] = 0 \wedge 1 \leq p \leq y_2| \wedge y_2 \leq n-1 \rightarrow (\exists k)(y_2+1 \leq k \leq n)\, a[k] = 0. \tag{21}$$

Podľa axiómy pre priraďovací príkaz dostávame

$$\{S(y_1, y_2, y_3, a) \wedge y_3 = b[y_1]\}\, y_2 := y_2{+}1;\, \{b[y_1] = |\{p : a[p] = 0 \wedge 1 \leq p < y_2|\} \tag{22}$$

Z platnosti (21) vyplýva, že existuje hodnota k', ktorá spĺňuje (23)

$$k' = \min\{k : y_2 \leq k \leq n \wedge a[k] = 0\} \tag{23}$$

Označme

$$T(y_2, a) : y_2 \leq k' \wedge b[y_1] = |\{p : a[p] = 0 \wedge 1 \leq p < y_2| \tag{24}$$

$$F3 : N \to N, \text{ t. j. } F3(y_2) = k' - y_2.$$

Funkcia $F3$ e klesajúca a nadobúda hodnoty z fundovanej množiny $(N, <)$.
Aplikáciou pravidla pre príkaz cyklu dostávame

$$\{T([y_2 + 1 \sharp y_2], a) \wedge a[y_2 + 1 \sharp y_2] \neq 0\}$$
$$y_2 := y_2 + 1 \{T(y_2, a) \wedge F3(y_2) < F3(y_2 + 1)\}$$

$$\{T(y_2, a)\} \text{ while } a[y_2] \neq 0 \text{ do}$$

$$y_2 := y + 2 + 1 \{T(y_2, a) \wedge a[y_2] = 0\}. \tag{25}$$

Z (23) a (25) vyplýva $y_2 = k'$.

Aplikáciou pravidla pre kompozíciu príkazov na (22) a (25) dostávame
$\{S(y_1, y_2, y_3, a) \wedge y_3 = b[y_1]\}$
$y_2 := y_2 + 1;$
while $a[y_2] \neq 0$ do $y_2 := y_2 + 1$ od

$$\{T(y_2, a) \wedge a[y_2] = 0\}. \tag{26}$$

Zrejme platí

$$\{T(y_2, a) \wedge a[y_2] = 0\}\, a[y_2] := y_1\, \{T(y_2, a) \wedge a[y_2] = y_1\}. \tag{27}$$

Podľa axiómy pre priraďovací príkaz platí
$\{S([y_1 + 1 \sharp y_1], [0 \sharp y_2], [0 \sharp y_3], a)\}$

$$(y_1, y_2, y_3) := (y_1 + 1, 0, 0)\{S(y_1, y_2, y_3, a)\} \tag{28}$$

Z (28), (26) a (27) a pravidla pre kompozíciu príkazov dostávame

$\{S(y_1 + 1, 0, 0, a)\}$
$(y_1, y_2, y_3) := (y_1 + 1, 0, 0);$
while $y_3 < b[y1]$ do
$y_2 := y_2 + 1;$
if $a[y_2] = 0$ then $y_3 := y_3 + 1$ fi;
od;
$y_2 := y_2 + 1;$
while $a[y_2] \neq 0$ do $y_2 := y_2 + 1$ od
$a[y_2] := y_1;$
$\{T(y_2, a) \wedge a[y_2] = y_1\}$

Zrejme platí

$$T(0, a) \equiv S(y_1 + 1, 0, 0, a). \tag{29}$$

Aplikáciou pravidla pre príkaz cyklu na posledný induktívny výraz a (29) a použitím funkcie $F4 : N^+ \to N^+$, t. j. $F4(y_1) = n - y_1$ dostávame výsledok.
Tým je vlastne induktívny výraz, ktorý predstavuje náš program so špecifikáciami predstavujúcimi vstupnú a výstupnú podmienku.