KATEDRA INFORMATIKY
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY
UNIVERZITA KOMENSKÉHO, BRATISLAVA

VERIFIKÁCIA PROGRAMOV

Bakalárska práca

Ján Kliman

študijný odbor: 9.2.1. informatika
vedúci: RNDr. Borislav Šuster, Csc

Bratislava, 2007

Čestné prehlásenie

Prehlasujem, že som túto prácu napísal sám, s použitím uvedenej literatúry, a s využitím teoretických poznatkov získaných počas môjho štúdia, a potvrdzujem to svojím podpisom

Bratislava, jún 2007

Ján Kliman .....

Poďakovanie

Ďakujem hlavne svojím rodičom, ktorí mi umožnili štúdium na vysokej škole, ďakujem aj za neustálu morálnu podporu počas štúdia od všetkých členov rodiny.

Ďakujem RNDr. Borislavovi Šusterovi za cenné rady a pripomienky, ktoré mi pomohli pri písaní tejto práce.

Abstrakt

Táto práca podáva prehľad o Hoareovej metóde, ktorá sa používa na dôkaz čiastočnej korektnosti programov, a o metóde dobre fundovaných množín, ktorá sa používa na dôkaz konečnosti programu. Tieto teoretické postupy sú následne aplikované na dôkaz čiastočnej správnosti programu a na dôkaz, že tento program skončí.

Kľúčové slová : správnosť programov, verifikácia programov, Hoareova metóda

Cieľ práce

Verifikácia programov je vo všeobecnosti algoritmicky neriešiteľná úloha. Keďže neexistuje algoritmus, ktorý verifikuje každý program pre každý vstupný a výstupný predikát, verifikácia programov je umenie, rovnako ako hudba či matematika.

Ukázalo sa, že overovanie správnosti algoritmov pomocou testovania je značne nespolahlivý postup, pretože množina prípustných vstupov môže byť nekonečná, takže nie je možné overiť každý vstup programu. Naproti tomu, overovanie pomocou verifikácie umožňuje skontrolovať všetky prípustné vstupy programu.

Ďalej sa ukázalo, že programátor, ktorý ovláda umenie verifikácie programov, s vyššou pravdepodobnosťou napíše bezchybný program, ako programátor, ktorý toto umenie neovláda.

Cieľom tejto práce je naučiť sa základom tohto verifikačného umenia tým, že sa podá prehľad o základných verifikačných technikách, ktoré sa následne aplikujú na verifikovanie programu.

Obsah

Čestné prehlásenie	1
Poďakovanie	2
Abstrakt	3
Cieľ práce	4
Obsah	5
Úvod	6
Štruktúrovaný program	7
Správnosť programov	9
Hoareova metóda	9
Metóda dobre fundovaných množín	12
Príklad	13
Špecifikácia	14
Návrh programu	15
Dôkaz čiastočnej správnosti programu	18
Dôkaz čiastočnej správnosti cyklu	18
Dôkaz čiastočnej správnosti príkazov vetvenia	24
Dôkaz čiastočnej správnosti programu	28
Dôkaz totálnej správnosti programu	29
Záver	31
Literatúra	32

Úvod

Informačno – komunikačná revolúcia prináša našej spoločnosti mnoho výhod, no aj množstvo problémov. Naše životy sú v čoraz väčšej miere závislé na čipoch, počítačoch, či na ich softvérovom vybavení. Moderné bankovníctvo, strojárska výroba, no už ani automobily sa nezaobídu bez výpočtovej techniky, ktorá spravuje stav našich financií vložených v banke alebo riadi činnosť strojov. Fakt, že sa naša spoločnosť stala vo veľkej miere závislá na výpočtovej technike, je nepopierateľný.

Jedno ľudové príslovie hovorí, že oheň je dobrý sluha, ale zlý pán. O výpočtovej technike platí to isté. Pokiaľ sa správa tak, ako to od nej očakávame, tak nám slúži a pomáha. Avšak nemusí to tak byť, hoci sú počítače a počítačové programy deterministické. Za všetkým sú totiž ľudia. A ľudia robia chyby. A keďže ľudia konštruujú počítače a píšu do nich programy, niektoré ich chyby môžu spôsobiť, že sa výpočtová technika začne správať tak, ako to od nej neočakávame. To, samozrejme, môže spôsobiť značné problémy.

Sú programy, pri ktorých nám chyby až tak veľmi neprekážajú. Zväčša sa objavujú náhodne pri praktickom používaní programov, a priebežne sa odstraňujú. (Samozrejme, bavíme sa o chybách, ktoré nezachytili testeri). No ak by sa podobná „malá chyba“ objavila v softvéri programu, ktorý kontroluje odpaľovanie jadrových hlavíc či chod elektrární, následky by mohli byť katastrofické. Preto je na mieste pýtať sa, ako sa dá podobným problémom zabrániť.

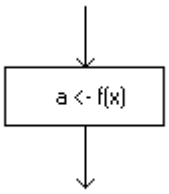
Jednou z možností je tvoriť verifikované programy. Teda vytvoriť špecifikáciu, podľa nej napísať program, a následne ho verifikovať, teda dokázať o ňom, že vždy skončí (nezasekne sa ani sa nezacyklí), a že pre každý vstup, ktorý spĺňa vstupnú podmienku špecifikácie, program vráti výstup, ktorý bude spĺňať výstupnú podmienku špecifikácie.

Prax ukázala, že je vhodné, aby verifikačné metódy ovládali samotní tvorcovia programov. Používanie týchto metód už pri návrhu programov výrazne znižuje chybovosť programov a v konečnom dôsledku šetrí čas a znižuje náklady.

Štruktúrovaný program

Cieľom tejto kapitoly je objasniť, čo je štruktúrovaný program, keďže táto práca pojednáva o Hoareovej metóde, a Hoareova metóda dokazuje čiastočnú správnosť len štruktúrovaných programov.

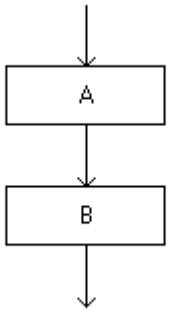
Definícia:
Príkaz priradenia premennej a priradí hodnotu $f(\overline{x})$ a zapisujeme ho $a \leftarrow f(\overline{x})$.



(schéma priraďovaceho príkazu)

Ďalej definujeme riadiace štruktúry programu.

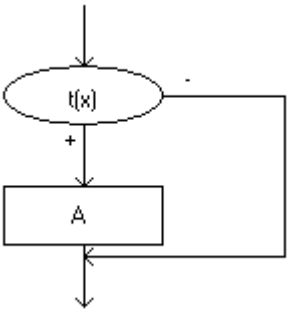
Definícia:
Zložený príkaz C je postupnosť príkazov, ktoré zapisujeme $A;B$, kde A, B sú ľubovoľné príkazy.



(schéma zloženého príkazu)

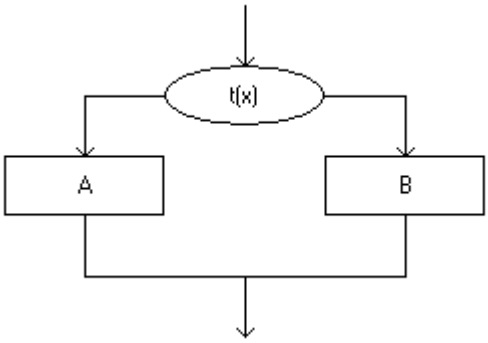
Definícia:
Podmienený príkaz má dve verzie.

a) Podmienený príkaz je príkaz, ktorý najskôr vyhodnotí logickú podmienku $t(\overline{x})$ a v prípade, že $t(\overline{x})$ má hodnotu true, vykoná príkaz A. Tento príkaz píšeme if $t(\overline{x})$ then A .



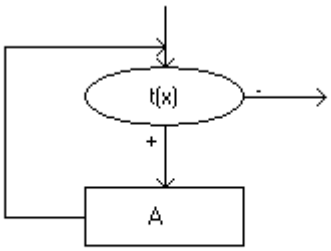
(schéma prvej verzie podmieneného príkazu)

b) Podmienený príkaz je príkaz, ktorý najskôr vyhodnotí logickú podmienku $t(\overline{x})$ a v prípade , že $t(\overline{x})$ má hodnotu true vykoná príkaz A a v prípade, že $t(\overline{x})$ má hodnotu false, vykoná príkaz B. Píšeme if $t(\overline{x})$ then A else B .



(schéma druhej verzie príkazu priradenia)

Definícia:
Príkaz cyklu najskôr vyhodnotí logickú podmienku $t(\overline{x})$ a potom vykonáva príkaz B pokým je logická podmienka $t(\overline{x})$ pravdivá.
Píšeme while $t(\overline{x})$ do A .



(schéma príkazu cyklu)

Definícia:
Štruktúrovaný program je

1. príkaz priradenia
2. ak A, B sú štruktúrované programy a $t(\bar{x})$ je logická podmienka, tak aj if $t(\bar{x})$ then A, if $t(\bar{x})$ then A else B, while $t(\bar{x})$ do A a $A;B$ je štruktúrovaný program.

Správnosť programov

Definícia:
Špecifikácia je dvojica predikátov $(\omega(\bar{x}), \psi(\bar{x}, \bar{z}))$, kde $\omega(\bar{x})$ je vstupný predikát a $\psi(\bar{x}, \bar{z})$ je výstupný predikát. Vstupný predikát kladie obmedzenia na hodnoty prvkov, ktoré môžu byť použité ako vstupné hodnoty programu, výstupný predikát popisuje vzťah, ktorý musia spĺňať výstupné hodnoty programu vzhľadom k vstupným po skončení výpočtu.

Definícia:
Program P končí pre ω, ak program P skončí pre všetky hodnoty $\bar{x}$, pre ktoré je $\omega(\bar{x})$ pravdivé.

Definícia:
Program P je čiastočne správny vzhľadom k ω, ψ, ak pre každú hodnotu $\bar{x}$, pre ktorú P končí a $\omega(\bar{x})$ je pravdivý, je pravdivý aj $\psi(\bar{x}, P(\bar{x}))$. Uvedenú definíciu vyjadrujeme aj formulou

$$(\forall \bar{x})(P \text{ skončí pre } \bar{x} \wedge \omega(\bar{x}) \Rightarrow \psi(\bar{x}, P(\bar{x})))$$

Definícia:
Program P je totálne správny vzhľadom k ω, ψ, ak pre každú hodnotu $\bar{x}$, pre ktorú je $\omega(\bar{x})$ pravdivý, P končí a súčasne je pravdivý $\psi(\bar{x}, P(\bar{x}))$, čo môžeme vyjadriť formulou

$$(\forall \bar{x})(\omega(\bar{x}) \Rightarrow P \text{ skončí pre } \bar{x} \wedge \psi(\bar{x}, P(\bar{x})))$$ [2].

Hoareova metóda

Na dôkaz čiastočnej správnosti môjho programu použijem Hoareovu metódu, ktorú zaviedol sir Charles Antony Richard Hoare ([1]).

Sir Hoare je významný britský vedec, ktorý priniesol informatickej vede tri významné objavy. Prvým z nich bol algoritmus na triedenie prvkov známy ako Quicksort, ktorý je aj dnes najrozšírenejším triediacim algoritmom. Druhým významným objavom bola Hoareova logika, ktorú použijem v svojom dôkaze čiastočnej korektnosti, a jeho tretím významným prínosom je formálny jazyk CSP, ktorým možno špecifikovať vzájomné správanie sa súbežných procesov (napr. problém obedujúcich filozofov).

Hoareova metóda dôkazu čiastočnej korektnosti používa tzv. induktívne výrazy, ktoré zapisujeme v tvare

$$\{p(\bar{x},\bar{y})\}B\{q(\bar{x},\bar{y})\},$$

kde p a q sú predikáty a B je segment programu. Tento zápis znamená, že ak platí v okamihu pred vykonaním časti programu B predikát $p(\bar{x},\bar{y})$ pre hodnoty $\bar{x},\bar{y}$ a B končí, tak po vykonaní B bude pre hodnoty $\bar{x},\bar{y}$ platiť predikát $q(\bar{x},\bar{y})$. Ak teda chceme dokázať čiastočnú korektnosť celého programu P Hoareovou metódou, musíme odvodiť induktívny výraz

$$\{p(\bar{x},\bar{y})\}P\{q(\bar{x},\bar{y})\}.$$

Tu sa žiada povedať, že hodnoty premenných $\bar{y}$ môžu byť iné v predikáte p a iné v predikáte q. Je to preto, lebo hodnoty $\bar{y}$ môžu byť modifikované nejakou časťou programu P.

Na dôkaz čiastočnej správnosti programu P slúžia verifikačné pravidlá, ktoré sú tvorené axiomou priradenia a odvodzovacími pravidlami pre riadiace štruktúry.

Axioma priradenia je induktívny výraz tvaru

$$\{p(\bar{x},g(\bar{x},\bar{y}))\}\bar{y}\leftarrow g(\bar{x},\bar{y})\{p(\bar{x},\bar{y})\}. \qquad (VP_1)$$

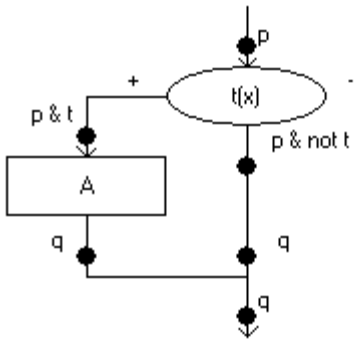
Tento induktívny výraz znamená, že ak tesne pred vykonaním priradenia platí predikát p pre hodnoty $\bar{x},g(\bar{x},\bar{y})$, po vykonaní príkazu priradenia bude tento predikát p platiť pre hodnoty $\bar{x},\bar{y}$.

Ďalšími verifikačnými pravidlami sú pravidlá vetvenia. Keďže máme dva tvary podmieneného príkazu (alebo tiež nazývaného príkaz vetvenia), máme aj dva tvary pravidla vetvenia.

a) Prvý tvar pravidla vetvenia je tvaru

$$\frac{\{p\wedge t\}A\{q\}\ a\ (p\wedge \text{not } t)\Rightarrow(q)}{\{p\}\ \text{if } t \text{ then } A\ \{q\}} \quad (VP_2).$$

Prvý tvar pravidla podmienky hovorí, že ak pred vykonaním príkazu A platí predikát $p\wedge t$ a po jeho vykonaní platí q, a súčasne je pravdivá formula $(p\wedge \text{not } t)\Rightarrow q$, tak ak pred vykonaním príkazu if t then A platí predikát p, po jeho vykonaní bude platiť q. Formulu $(p\wedge \text{not } t)\Rightarrow q$ chápeme ako uzavretú.

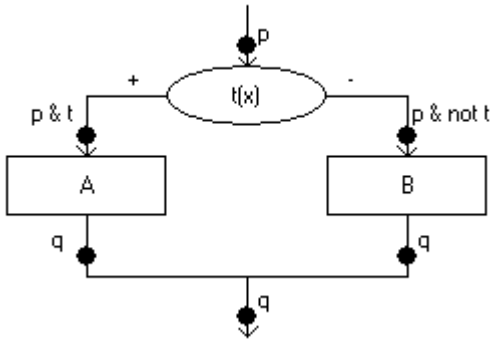


(schéma prvého tvaru pravidla vetvenia)

b) Druhý tvar pravidla podmienky je

$$\frac{\{p \wedge t\}A\{q\} \text{ a } \{p \wedge \text{not } t\}B\{q\}}{\{p\} \text{ if } t \text{ do } A \text{ else } B\{q\}} \quad (VP_3) \ .$$

Jeho význam je ten, že ak pred vykonaním príkazu A platí predikát $p \wedge t$ a po jeho vykonaní platí q, a pred vykonaním príkazu B platí predikát $p \wedge \text{not } t$ a po jeho vykonaní platí q, znamená to, že ak pred vykonaním príkazu $\text{if } t \text{ then A else B}$ platí predikát p, po vykonaní príkazu $\text{if } t \text{ then A else B}$ bude platiť predikát q.

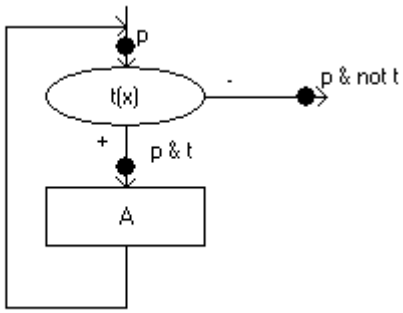


(schéma druhého tvaru pravidla vetvenia)

Pre dôkaz čiastočnej korektnosti programu sa používa aj pravidlo cyklu, ktoré zapisujeme

$$\frac{\{p \wedge t\}A\{p\}}{\{p\} \text{ while } t \text{ do } A\{p \wedge \text{not } t\}} \quad (VP_4) \ .$$

Tento zápis znamená, že ak platí induktívny výraz $\{p \wedge t\}A\{p\}$, platí aj induktívny výraz $\{p\} \text{ while } t \text{ do } A \{p \wedge \text{not } t\}$. Pravidlo sa opiera o skutočnosť, že pokiaľ príkaz A zachováva platnosť predikátu p, predikát p bude platiť aj po ľubovoľnom počte za sebou idúcich vykonaní príkazu A.



(schéma pravidla cyklu)

Ďalšie pravidlo je pravidlo sekvencie pre zložený príkaz.

$$\frac{\{p\}A\{q\} \text{ a } \{q\}B\{r\}}{\{p\}A;B\{r\}} \quad (VP_5)$$

Toto pravidlo sa používa na zreťazovanie príkazov. Jeho významom je, že pokiaľ má príkaz A rovnakú výstupnú podmienku ako má príkaz B vstupnú, tak potom je induktívny výraz $\{p\}A;B\{r\}$ pravdivý.

Posledné spomenuté pravidlo je pravidlo konsekvencie, ktoré má dve verzie.

Prvé pravidlo konsekvencie je zosilnením vstupnej podmienky

$$\frac{p \Rightarrow q \ a \ \{q\}A\{r\}}{\{p\}A\{r\}} \quad (VP_6)$$

a druhé umožňuje zoslabiť výstupnú podmienku

$$\frac{\{p\}A\{q\} \ a \ q \Rightarrow r}{\{p\}A\{r\}} \quad (VP_7) \ .$$

Veta: (Metóda verifikačných pravidiel – Hoare)

Nech je daný štruktúrovaný program P, vstupný predikát $\omega(\bar{x})$ a výstupný predikát $\psi(\bar{x},\bar{z})$. Ak je možné postupnou aplikáciou uvedených verifikačných pravidiel odvodiť induktívnu formulu $\omega(\bar{x})P\psi(\bar{x},\bar{z})$, potom je program P čiastočne korektný vzhľadom k vstupnému predikátu ω a výstupnému predikátu ψ. [2]

Dôkaz vety presahuje cieľ tejto práce.

Metóda dobre fundovaných množín

Definícia:

Čiastočne usporiadaná množina (W,<) je dvojica pozostávajúca z neprázdnej množiny W a binárnej relácie <, ktorá je ostrým čiastočným usporiadaním (relácia < je ireflexívna, asymetrická a tranzitívna).

Definícia:

Čiastočne usporiadanú množinu (W,<) nazveme dobre fundovaná, ak neobsahuje žiadnu nekonečne klesajúcu postupnosť.

Dôkaz ukončenia programu Floydovou metódou dobre fundovaných množín využíva deliace body. Tie sa zvolia tak, aby každý cyklus obsahoval aspoň jeden deliaci bod. Ku každému deliacemu bodu i priradíme induktívnu podmienku $q_i(\bar{x},\bar{y})$ tak, aby platilo

a) pre každú cestu z východzieho bodu k deliacemu bodu j, ktorá neprechádza iným deliacim bodom, platí

$$\forall \bar{x}[\omega(\bar{x}) \wedge R_\alpha(\bar{x}) \Rightarrow q_j(\bar{x}, r_\alpha(\bar{x}))] \quad (T_1)$$

b) pre každú cestu α z deliaceho bodu i do deliaceho bodu j, ktorá neprechádza iným deliacim bodom, platí

$$\forall \bar{x} \forall \bar{y}[q_i(\bar{x},\bar{y}) \wedge R_\alpha(\bar{x},\bar{y}) \Rightarrow q_j(\bar{x}, r_\alpha(\bar{x},\bar{y}))] \quad (T_2)$$

$R_\alpha(\bar{x},\bar{y})$ je podmienka nato, aby program P prešiel cestu z i do j

$r_\alpha(\bar{x},\bar{y})$ je označenie pre popis zmien hodnôt vektoru $\bar{y}$ spôsobenou príkazmi v ceste α.

Druhým krokom je voľba vhodnej dobre fundovanej množiny (W,<) a priradenie funkcie $u_i(\bar{x},\bar{y})$ každému deliacemu bodu i. Táto funkcia zobrazuje vektory premenných $(\bar{x},\bar{y})$ do množiny W. Pre funkciu $u_i(\bar{x},\bar{y})$ musí platiť

$$\forall \bar{x} \forall \bar{y} [q_i(\bar{x},\bar{y}) \Rightarrow u_i(\bar{x},\bar{y}) \in W] \quad (T_3).$$

V poslednom kroku dokážeme platnosť podmienky ukončenia, tj. že pre každú cestu α z deliaceho bodu i do deliaceho bodu j, ktorá neprechádza iným deliacim bodom, platí

$$\forall \bar{x} \forall \bar{y} \{q_i(\bar{x},\bar{y}) \wedge R_\alpha(\bar{x},\bar{y}) \Rightarrow [u_i(\bar{x},\bar{y}) > u_j(\bar{x}, r_\alpha(\bar{x},\bar{y}))]\} \quad (T_4),$$

čo znamená, že pri prechode z deliaceho bodu i do deliaceho bodu j nastane ostrý pokles hodnôt z dobre fundovanej množiny W.

Veta: (Metóda dobre fundovaných množín – Floyd)

Nech je daný program P a vstupný predikát $\omega(\bar{x})$ a je možné zostrojiť postupne deliace body všetkých cyklov programu P, vhodné dobré induktívne podmienky, vhodnú dobre fundovanú množinu a dobré funkcie a podmienky ukončenia. Ak sú všetky podmienky ukončenia pravdivé, program P končí pre ω. Dobrými induktívnymi podmienkami a dobrými funkciami rozumieme tie, ktoré spĺňajú podmienky T_1, T_2, T_3, T_4. [2]

Príklad

Teraz pristúpime k dôkazu totálnej správnosti konkrétneho programu. Nato budeme musieť zo slovného zadania vytvoriť špecifikáciu, následne napísať program a dokázať jeho totálnu správnosť. Dôkaz čiastočnej správnosti vykonáme Hoareovou metódou, dôkaz zastavenia vykonáme Floydovou metódou dobre fundovaných množín.

Zadanie príkladu:

Majme súbor f obsahujúci postupnosť reálnych čísel. Dĺžka súboru nie je informáciou pre programátora. Tento súbor bude vstupom programu, ktorý ako svoj výstup vráti počet takých po sebe idúcich čísel, ktoré majú vlastnosť $V: R \times R \times R \to \{False, True\}$. Ako príklad môže slúžiť vlastnosť lokálneho maxima $V(x_1, x_2, x_3) = (x_1 < x_2 > x_3)$.

Definícia

Neprázdny súbor f nastavený na čítanie je dvojica $f = ((f_1, f_2, ..., f_n), F)$, kde f_i je i-ty prvok v súbore f, $i = \overline{(1,n)}$, a F je pozícia čítacej hlavy, $F \in \mathbb{N}$. Veľkosť súboru označíme $dl(f) = n$. Prázdny súbor budeme označovať $f = ((), F)$ a jeho veľkosť $dl(f) = 0$.

Definícia:
Boolovská funkcia $EOF(f)=(F>dl(f))$. Funkcia vracia hodnotu true, ak sa čítacia hlava dostala na koniec súboru a už nie je možné načítať z neho ďalší prvok; inak vracia hodnotu false.

Definícia:
Príkaz $Reset(f)$ dostane na vstup súbor $f=((f_1,f_2,...,f_n),F)$ alebo $f=((),F)$ kde $F\in\mathbb{N}$ a vráti súbor $f=((f_1,f_2,...,f_n),1)$ ak pôvodný f je neprázdny alebo $f=((),1)$ ak je prázdny.
Táto funkcia nastaví čítaciu hlavu na hodnotu 1, pritom ju nezaujíma, či je súbor neprázdny. Príkaz $Reset(f)$ je teda ekvivalentný príkazu $F\leftarrow 1$.

Definícia:
Nech f je neprázdny súbor $((f_1,f_2,...,f_n),F)$ a nech $F\leq n$. Príkaz $a\leftarrow Read(f)$ priradí premennej a prvok f_F súboru f a F priradí hodnotu $F+1$. Teda príkaz $a\leftarrow Read(f)$ načíta hodnotu a so súboru f a súčasne posunie čítaciu hlavu. Príkaz $a\leftarrow Read(f)$ je teda ekvivalentný zloženému príkazu $a\leftarrow f_F; F\leftarrow F+1$. Pre prázdny súbor nie je príkaz $a\leftarrow Read(f)$ definovaný.

Špecifikácia

Najskôr je potrebné na základe slovného zadania vytvoriť špecifikáciu programu, teda vstupný a výstupný predikát. Každý vstup musí spĺňať vstupný predikát $\omega(f)$ a každý výstup musí spĺňať výstupný predikát $\psi(f,z)$. Podľa zadania mám na vstupe súbor reálnych čísel, ktorý môže byť prázdny. Vstupný predikát programu je

$$\omega(f)=(f=((f_1,f_2,...,f_n),F)\wedge n\in\mathbb{N}\setminus\{0\}\wedge F\in\mathbb{N}\wedge(\forall i)(1\leq i\leq n\Rightarrow f_i\in\mathbb{R})\ \vee$$
$$f=((),F)\wedge F\in\mathbb{N}).$$

Výstupom programu má byť číslo z rovné počtu takých po sebe idúcich trojíc, ktoré spĺňajú vlastnosť V. Čiže vo výstupnej premennej, označme ju z, sa po prečítaní celého súboru bude nachádzať počet týchto trojíc. Výstupný predikát programu je

$$\psi(f,z)=(\omega(f)\wedge dl(f)<F\wedge z=|\{i\mid V(f_{i-2},f_{i-1},f_i)\wedge f_{i-2}\in f\wedge f_{i-1}\in f\wedge f_i\in f\wedge i\in\mathbb{N}\}|).$$

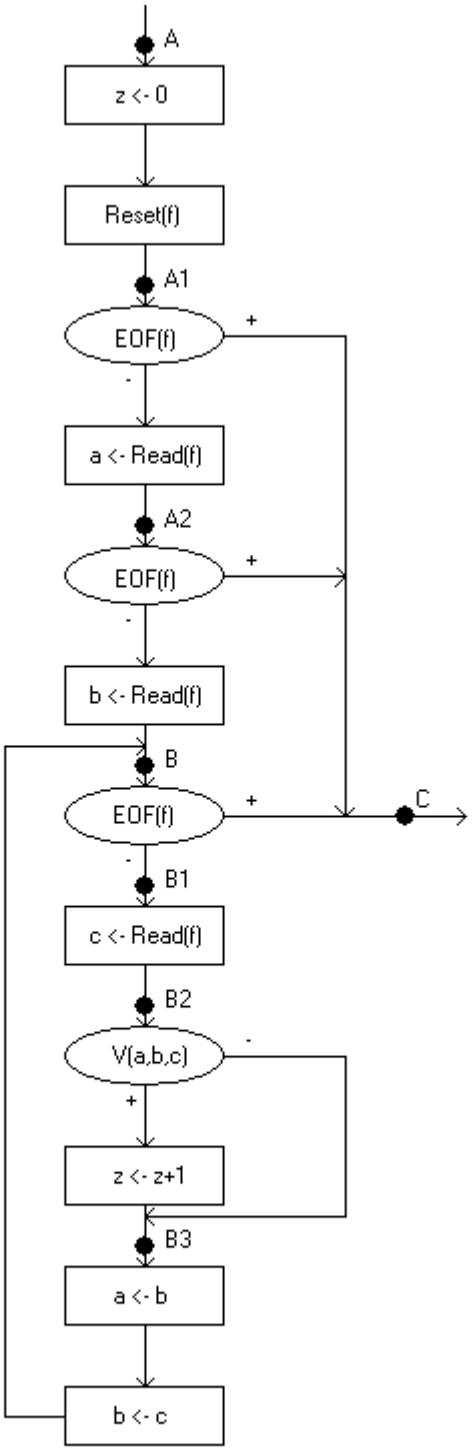
Návrh programu

Navrhneme program P, ktorý spĺňa podmienku $\{\omega(f)\}\ P\ \{\psi(f,z)\}$ na čiastočnú správnosť a skončí pre $\omega(f)$. Program vyzerá nasledovne:

```
z←0 ;
Reset(f) ;
if not EOF(f)
   then begin
      a←Read(f) ;
      if not EOF(f)
      then begin
         b←Read(f) ;
         while not EOF(f)
         begin
            c←Read(f) ;
            if V(a,b,c)
               then z←z+1 ;
            a←b ;
            b←c ;
         end
      end
   end
```

Begin a end označujú začiatok a koniec zloženého príkazu.

Tento program je štruktúrovaný, čo lepšie vidieť z nasledujúceho vývojového diagramu.



(vývojový diagram programu)

Texty v obdĺžnikoch sú príkazmi, texty v elipsách sú testy. Šípkami je znázornená postupnosť príkazov. Zo všetkých testov vychádzajú dve šípky, jedna je označená znamienkom +,

druhá je označená -. Význam tohto je taký, že pokiaľ test vráti hodnotu true, program pokračuje ďalej po ceste označenej plusom, pokiaľ test vráti hodnotu false, beh programu pokračuje po ceste označenej mínusom.

V uvedenom vývojovom diagrame sa nachádzajú aj tri body, ktoré som označil písmenami A, B, C. Tieto body sa nazývajú deliacimi bodmi programu a majú veľký význam pri dokazovaní čiastočnej správnosti programu Floydovou metódou induktívnych podmienok, a tiež pri dôkaze, že program skončí. Bodu A sa priradí vstupný predikát špecifikácie, bodu C výstupný. Bodu B sa priradí tzv. invariant - predikát, ktorý platí vždy, keď sa program prechádza tým miestom. Body A1, A2, B1, B2 a B3 sú deliace body, ktoré pomôžu sprehľadniť dôkaz čiastočnej správnosti programu.

Príkazy obsahujúce Reset, Read a EOF nahradíme testami a priradeniami, ktoré pracujú s celočíselnými premennými a, b, c, F.

```
z←0 ;
F←1 ;
  if F≤dl(f)
     then begin
        a←f_F ;
        F←F+1 ;
        if F≤dl(f)
           then begin
           b←f_F ;
           F←F+1 ;
           while F≤dl(f)
              begin
              c←f_F ;
              F←F+1 ;
              if V(a,b,c)
                 then z←z+1 ;
              a←b ;
              b←c ;
              end
        end
     end
```

Takto prepísaný program je ekvivalentný pôvodnému programu.

Dôkaz čiastočnej správnosti programu

Na dôkaz čiastočnej správnosti programu P použijem Hoarreovu metódu. Je to metóda dôkazu zdola nahor, teda najskôr sa dokáže čiastočná správnosť elementárnych príkazov priradenia, a následne sa z predpokladu čiastočnej správnosti nejakých príkazov dokáže čiastočná správnosť zložitejších príkazov vetvenia, cyklu či zložených príkazov, a z toho nakoniec vyplynie čiastočná správnosť celého programu P.

Dôkaz čiastočnej správnosti cyklu

Najskôr dokážeme čiastočnú korektnosť cykla programu tým, že dokážeme čiastočnú korektnosť jednotlivých príkazov cyklu vzhľadom na nejaké dobre zvolené predikáty a použijeme pravidlo sekvencie. Vieme (viď Hoareova metóda), že ak telo cyklu zachováva platnosť nejakého predikátu, tento predikát bude platiť aj po niekoľkonásobnom vykovaní príkazov cyklu. Tento predikát nazvývame invariant. Za účelom definovania invariantnej formuly pre náš program budeme potrebovať formulu

$$z=|\{i|f_{i-2}\in f \wedge f_{i-1}\in f \wedge f_i\in f \wedge V(f_{i-2},f_{i-1},f_i) \wedge i\in \mathbb{N} \wedge i<F\}|,$$

ktorá hovorí, že z sa rovná počtu tých po sebe idúcich doteraz prečítaných prvkov súboru f, ktoré majú vlastnosť V. Množinu

$$\{i|f_{i-2}\in f \wedge f_{i-1}\in f \wedge f_i\in f \wedge V(f_{i-2},f_{i-1},f_i) \wedge i\in \mathbb{N} \wedge i<F\}$$

označíme M_F (pre skrátenie zápisu). Ďalej si zadefinujeme invariantnú formulu $I(f,z)$, o ktorej dokážeme, že bude platiť vždy, keď sa výpočet dostane do bodu B.

$$I(f,z) = (\omega(f) \wedge z=|M_F| \wedge 3\leqslant F\leqslant dl(f)+1 \wedge a=f_{F-2} \wedge b=f_{F-1}).$$

1. Predpokladajme, že v deliacom bode B invariantná formula $I(f,z)$ platí. Náš dôkaz začíname v bode B_1, kde platí formula $I(f,z) \wedge F\leqslant dl(f)$, čo je ekvivalentné formule

$$(\omega(f) \wedge z=|M_F| \wedge 3\leqslant F\leqslant dl(f) \wedge a=f_{F-2} \wedge b=f_{F-1}).$$

Očakávame, že po vykonaní príkazu $c\leftarrow Read(f)$ bude platiť formula

$$(\omega(f) \wedge z=|M_{F-1}| \wedge 4\leqslant F\leqslant dl(f)+1 \wedge a=f_{F-3} \wedge b=f_{F-2} \wedge c=f_{F-1}).$$

Naše tvrdenie môžeme formálne zapísať

$$\{(\omega(f) \wedge z=|M_F| \wedge 3\leqslant F\leqslant dl(f) \wedge a=f_{F-2} \wedge b=f_{F-1})\}$$
$$c\leftarrow Read(f)$$
$$\{(\omega(f) \wedge z=|M_{F-1}| \wedge 4\leqslant F\leqslant dl(f)+1 \wedge a=f_{F-3} \wedge b=f_{F-2} \wedge c=f_{F-1})\}$$

Dôkaz bodu 1. vykonáme v niekoľkých krokoch. Pripomíname, že príkaz $c\leftarrow Read(f)$ je zložený príkaz $c\leftarrow f_F; F\leftarrow F+1$

a) najskôr zosilníme vstupnú podmienku. Formula

$$(\omega(f) \wedge z=|M_F| \wedge 3 \leqslant F \leqslant dl(f) \wedge a=f_{F-2} \wedge b=f_{F-1})$$
$$\Rightarrow$$
$$(\omega(f) \wedge z=|M_F| \wedge 3 \leqslant F \leqslant dl(f) \wedge a=f_{F-2} \wedge b=f_{F-1} \wedge f_F=f_F)$$

je zjavne pravdivá.

b) použijeme axiomu priradenia

$$\{(\omega(f) \wedge z=|M_F| \wedge 3 \leqslant F \leqslant dl(f) \wedge a=f_{F-2} \wedge b=f_{F-1} \wedge f_F=f_F)\}$$
$$c \leftarrow f_F$$
$$\{(\omega(f) \wedge z=|M_F| \wedge 3 \leqslant F \leqslant dl(f) \wedge a=f_{F-2} \wedge b=f_{F-1} \wedge c=f_F)\}$$

c) zoslabíme výstupnú podmienku bodu b). Implikácia

$$(\omega(f) \wedge z=|M_F| \wedge 3 \leqslant F \leqslant dl(f) \wedge a=f_{F-2} \wedge b=f_{F-1} \wedge c=f_F)$$
$$\Rightarrow$$
$$(\omega(f) \wedge z=|M_F| \wedge 4 \leqslant F+1 \leqslant dl(f)+1 \wedge a=f_{F-2} \wedge b=f_{F-1} \wedge c=f_F)$$

je pravdivá.

d) Opäť použijeme axiomu priradenia

$$\{(\omega(f) \wedge z=|M_F| \wedge 4 \leqslant F+1 \leqslant dl(f)+1 \wedge a=f_{F-2} \wedge b=f_{F-1} \wedge c=f_F)\}$$
$$F \leftarrow F+1$$
$$\{(\omega(f) \wedge z=|M_{F-1}| \wedge 4 \leqslant F \leqslant dl(f)+1 \wedge a=f_{F-3} \wedge b=f_{F-2} \wedge c=f_{F-1})\}$$

e) Body b) a d) dáme dohromady pravidlom sekvencie.

$$\{(\omega(f) \wedge z=|M_F| \wedge 3 \leqslant F \leqslant dl(f) \wedge a=f_{F-2} \wedge b=f_{F-1} \wedge f_F=f_F)\}$$
$$c \leftarrow f_F$$
$$F \leftarrow F+1$$
$$\{(\omega(f) \wedge z=|M_{F-1}| \wedge 4 \leqslant F \leqslant dl(f)+1 \; a=f_{F-3} \wedge b=f_{F-2} \wedge c=f_{F-1})\}$$

f) Na body a) a e) použijeme pravidlo konsekvancie

$$\{(\omega(f) \wedge z=|M_F| \wedge 3 \leqslant F \leqslant dl(f) \wedge a=f_{F-2} \wedge b=f_{F-1})\}$$
$$c \leftarrow f_F$$
$$F \leftarrow F+1$$
$$\{(\omega(f) \wedge z=|M_{F-1}| \wedge 4 \leqslant F \leqslant dl(f)+1 \wedge a=f_{F-3} \wedge b=f_{F-2} \wedge c=f_{F-1})\} \quad ,$$

čím je bod 1. dokázaný.

2. Nasleduje dôkaz čiastočnej korektnosti podmieneného príkazu. Pred vstupom do podmieneného príkazu platí v bode B2 výstupný predikát príkazu $c \leftarrow \mathrm{Read}(f)$, ktorý tým pádom považujeme za vstupný predikát dokazovaného príkazu vetvenia. Chceme dokázať nasledovné tvrdenie

$$\{(\omega(f) \wedge z = |M_{F-1}| \wedge 4 \leqslant F \leqslant dl(f)+1 \wedge a = f_{F-3} \wedge b = f_{F-2} \wedge c = f_{F-1})\}$$
$$\text{if } V(a,b,c)$$
$$\text{then } z \leftarrow z+1$$
$$\{(\omega(f) \wedge z = |M_F| \wedge 4 \leqslant F \leqslant dl(f)+1 \wedge a = f_{F-3} \wedge b = f_{F-2} \wedge c = f_{F-1})\}$$

Pravdivosť tvrdenia 2. vyplýva z pravidla vetvenia. Najskôr však musíme dokázať platnosť jeho predpokladov.

a) Nech platí podmienka $V(a,b,c)$. Potom program vykoná priradenie $z \leftarrow z+1$. Preto potrebujeme dokázať platnosť induktívnej podmienky

$$\{(\omega(f) \wedge z = |M_{F-1}| \wedge 4 \leqslant F \leqslant dl(f)+1 \wedge a = f_{F-3} \wedge b = f_{F-2} \wedge c = f_{F-1} \wedge V(a,b,c))\}$$
$$z \leftarrow z+1$$
$$\{(\omega(f) \wedge z = |M_F| \wedge 4 \leqslant F \leqslant dl(f)+1 \wedge a = f_{F-3} \wedge b = f_{F-2} \wedge c = f_{F-1})\}$$

Potrebujeme dokázať, že množina M_F obsahuje o jeden prvok viac ako množina M_{F-1}. Zjavne sa musí jednať o prvok $F-1$, ktorý podľa definície nepatrí do M_{F-1}, ale patri do M_F, pretože z predpokladu $a = f_{F-3} \wedge b = f_{F-2} \wedge c = f_{F-1} \wedge V(a,b,c)$ vyplýva platnosť $V(f_{F-3}, f_{F-2}, f_{F-1})$ a $f_{F-3}, f_{F-2}, f_{F-1}$ sú prvky súboru f.

b) Zostáva dokázať platnosť implikácie

$$(\omega(f) \wedge z = |M_{F-1}| \wedge 4 \leqslant F \leqslant dl(f)+1 \wedge a = f_{F-3} \wedge b = f_{F-2} \wedge c = f_{F-1} \wedge \text{not } V(a,b,c))$$
$$\Rightarrow$$
$$(\omega(f) \wedge z = |M_F| \wedge 4 \leqslant F \leqslant dl(f)+1 \wedge a = f_{F-3} \wedge b = f_{F-2} \wedge c = f_{F-1})$$

Stačí ukázať, že i F nepatrí M_F. Ak by patril, potom by musela platiť podmienka $V(a,b,c)$, čo však nie je pravda.

Pravdivosť tvrdenia 2. teda vyplýva z tvrdení a) a b) a pravidla vetvenia.

3. Použitím pravidla sekvencie na body 1, a 2. dostávame pravdivý výraz

$$\{(\omega(f) \wedge z = |M_F| \wedge 3 \leqslant F \leqslant dl(f) \wedge a = f_{F-2} \wedge b = f_{F-1})\}$$
$$c \leftarrow \mathrm{Read}(f);$$
$$\text{if } V(a,b,c)$$
$$\text{then } z \leftarrow z+1$$
$$\{(\omega(f) \wedge z = |M_F| \wedge 4 \leqslant F \leqslant dl(f)+1 \wedge a = f_{F-3} \wedge b = f_{F-2} \wedge c = f_{F-1})\}$$

Dostali sme sa do bodu B3, v ktorom platí

$$(\omega(f) \wedge z=|M_F| \wedge 4 \leqslant F \leqslant dl(f)+1\ a=f_{F-3} \wedge b=f_{F-2} \wedge c=f_{F-1}).$$

V ďalších dvoch krokoch zistíme, aká formula bude platiť po vykonaní príkazov $a \leftarrow b$ a $b \leftarrow c$.

4. $\{(\omega(f) \wedge z=|M_F| \wedge 4 \leqslant F \leqslant dl(f)+1 \wedge a=f_{F-3} \wedge b=f_{F-2} \wedge c=f_{F-1})\}$

$a \leftarrow b$

$\{(\omega(f) \wedge z=|M_F| \wedge 4 \leqslant F \leqslant dl(f)+1 \wedge a=f_{F-2} \wedge c=f_{F-1})\}$

Pravdivosť uvedeného induktívneho výrazu vyplýva z pravdivosti implikácie

$$(\omega(f) \wedge z=|M_F| \wedge 4 \leqslant F \leqslant dl(f)+1 \wedge;\ a=f_{F-3} \wedge b=f_{F-2} \wedge c=f_{F-1})$$
$$\Rightarrow$$
$$(\omega(f) \wedge z=|M_F| \wedge 4 \leqslant F \leqslant dl(f)+1 \wedge b=f_{F-2} \wedge c=f_{F-1})$$

a z axiomy priradenia, ktorá je tvaru

$\{(\omega(f) \wedge z=|M_F| \wedge 4 \leqslant F \leqslant dl(f)+1 \wedge b=f_{F-2} \wedge c=f_{F-1})\}$

$a \leftarrow b$

$\{(\omega(f) \wedge z=|M_F| \wedge 4 \leqslant F \leqslant dl(f)+1 \wedge a=f_{F-2} \wedge c=f_{F-1})\}$

5. $\{(\omega(f) \wedge z=|M_F| \wedge 4 \leqslant F \leqslant dl(f)+1 \wedge a=f_{F-2} \wedge c=f_{F-1})\}$

$b \leftarrow c$

$\{(\omega(f) \wedge z=|M_F| \wedge 4 \leqslant F \leqslant dl(f)+1 \wedge a=f_{F-2} \wedge b=f_{F-1})\}$

Toto tvrdenie je pravdivé z tých istých dôvodov ako tvrdenie z bodu 4.

Pravdivosť nasledujúcich tvrdení je dôsledkom bodov 3., 4., a 5. a pravidla sekvencie.

6. $\{(\omega(f) \wedge z=|M_F| \wedge 3 \leqslant F \leqslant dl(f) \wedge a=f_{F-2} \wedge b=f_{F-1})\}$

$c \leftarrow Read(f)$;

if $V(a,b,c)$

then $z \leftarrow z+1$

$a \leftarrow b$

$\{(\omega(f) \wedge z=|M_F| \wedge 4 \leqslant F \leqslant dl(f)+1 \wedge a=f_{F-2} \wedge c=f_{F-1})\}$

7. $\{(\omega(f) \wedge z=|M_F| \wedge 3 \leqslant F \leqslant dl(f) \wedge a=f_{F-2} \wedge b=f_{F-1})\}$

$c \leftarrow Read(f)$

if $V(a,b,c)$

then $z \leftarrow z+1$;

$a \leftarrow b$;

$b \leftarrow c$

$\{(\omega(f) \wedge z=|M_F| \wedge 4 \leqslant F \leqslant dl(f)+1 \wedge a=f_{F-2} \wedge b=f_{F-1})\}$

Podarilo sa nám dokázať, že ak v bode B1 platí predikát

$$\{(\omega(f) \wedge z=|M_F| \wedge 3 \leqslant F \leqslant dl(f) \wedge a=f_{F-2} \wedge b=f_{F-1})\}$$

v bode B bude platiť predikát

$$\{(\omega(f) \wedge z=|M_F| \wedge 4 \leqslant F \leqslant dl(f)+1 \wedge a=f_{F-2} \wedge b=f_{F-1})\}$$

(viď vývojový diagram programu na strane 16)

8. Chceme dokázať čiastočnú korektnosť cyklu vzhľadom na invariant $I(f,z)$. Teda chceme dokázať platnosť

$\{\omega(f) \wedge z=|M_F| \wedge 3 \leqslant F \leqslant dl(f)+1 \wedge a=f_{F-2} \wedge b=f_{F-1}\}$

while not $EOF(f)$

begin

$c \leftarrow Read(f)$;

if $V(a,b,c)$

then $z \leftarrow z+1$;

$a \leftarrow b$;

$b \leftarrow c$;

end

$\{\omega(f) \wedge z=|M_F| \wedge 3 \leqslant F \leqslant dl(f)+1 \wedge a=f_{F-2} \wedge b=f_{F-1} \wedge F>dl(f)\}$

Podľa pravidla cyklu potrebujeme dokázať pravdivosť

$\{\omega(f) \wedge z=|M_F| \wedge 3 \leqslant F \leqslant dl(f) \wedge a=f_{F-2} \wedge b=f_{F-1}\}$

$c \leftarrow Read(f)$;

if $V(a,b,c)$

then $z \leftarrow z+1$;

$a \leftarrow b$;

$b \leftarrow c$;

$\{\omega(f) \wedge z=|M_F| \wedge 3 \leqslant F \leqslant dl(f)+1 \wedge a=f_{F-2} \wedge b=f_{F-1}\}$

Na dokázanie tvrdenia nám podľa bodu 7. stačí platnosť implikácie

$$(\omega(f) \wedge z=|M_F| \wedge 4 \leqslant F \leqslant dl(f)+1 \wedge a=f_{F-2} \wedge b=f_{F-1})$$
$$\Rightarrow$$
$$(\omega(f) \wedge z=|M_F| \wedge 3 \leqslant F \leqslant dl(f)+1 \wedge a=f_{F-2} \wedge b=f_{F-1})$$

ktorá vyplýva z pravdivosti tvrdenia $4 \leqslant F \Rightarrow 3 \leqslant F$.

Podarilo sa nám teda dokázať, že cyklus programu P je čiastočne korektný vzhľadom na invariant $I(f,z)$.

9. V tomto kroku ukážeme platnosť implikácie (po skončení cyklu)

$$(\omega(f) \wedge z=|M_F| \wedge 3 \leqslant F \leqslant dl(f)+1 \wedge a=f_{F-2} \wedge b=f_{F-1} \wedge F>dl(f))$$
$$\Rightarrow$$
$$(\omega(f) \wedge dl(f)<F \wedge z=|\{i \mid V(f_{i-2},f_{i-1},f_i) \wedge f_{i-2} \in f \wedge f_{i-1} \in f \wedge f_i \in f \wedge i \in \mathbb{N}\}|)$$

ktorá hovorí, že ak program opustí cyklus, potom spĺňa výstupnú podmienku špecikikácie. Množinu $\{i \mid V(f_{i-2},f_{i-1},f_i) \wedge f_{i-2} \in f \wedge f_{i-1} \in f \wedge f_i \in f \wedge i \in \mathbb{N}\}$ označíme písmenom K.

Pravdivosť formuly $\omega(f) \wedge F>dl(f)$ je zrejmá a ešte potrebujeme dokázať, že $|M_F|=|K|$. Inými slovami povedané treba dokázať, že tieto dve množiny majú rovnaké prvky.

Nech teda $j \in M_F$. To podľa definície M_F znamená, že

$$f_{j-2} \in f \wedge f_{j-1} \in f \wedge f_j \in f \wedge V(f_{j-2},f_{j-1},f_j) \wedge j \in \mathbb{N},$$ teda, že $j \in K$.

Na druhej stane, z predpokladu $j \notin M_F$, vyplýva

$$f_{j-2} \notin f \vee f_{j-1} \notin f \vee f_j \notin f \vee \text{not}\, V(f_{j-2},f_{j-1},f_j) \vee j \notin \mathbb{N} \vee j \geqslant m.$$

Ak platí $f_{j-2} \notin f \vee f_{j-1} \notin f \vee f_j \notin f \vee \text{not}\, V(f_{j-2},f_{j-1},f_j) \vee j \notin \mathbb{N}$, potom $j \notin K$.

Ak $j \geqslant F$, potom z predpokladu $\omega(f) \wedge 3 \leqslant F \leqslant dl(f)+1 \wedge F>dl(f)$ vyplýva $j>dl(f)$ a z toho $f_j \notin f$. Teda aj v tomto prípade $j \notin K$.

10. Z bodov 8. a 9.a pravidla sekvencie vyplýva

$\{I(f,z)\}$

```
while not EOF(f)
        begin
        c ← Read(f);
        if V(a,b,c)
                then z ← z+1;
        a ← b;
        b ← c;
        end
```

$\{\psi(f,z)\}$

Dôkaz čiastočnej správnosti príkazov vetvenia

Predpokladajme teraz, že sa náš program dostal do bodu A2, a že v ňom platí predikát

$$\omega(f) \wedge z=0 \wedge F=2 \wedge F \leqslant dl(f)+1 \wedge a=f_1 .$$

Potom program vyhodnotí test $F>dl(f)$. Ak platí, program skončí, ak nie, pokračuje príslušnou vetvou. Chceme teda dokázať nasledovný výraz

$\{\omega(f) \wedge z=0 \wedge F=2 \wedge F \leqslant dl(f)+1 \wedge a=f_1\}$

```
if not EOF(f)
    then begin
    b←Read(f);
    while not EOF(f)
        begin
        c←Read(f);
        if V(a,b,c)
            then z←z+1;
        a←b;
        b←c;
        end
    end
```

$\{\psi(f,z)\}$

Dôkaz prevedieme pravidlom vetvenia v nasledujúcich krokoch.

11. Predpokladajme, že test $F>dl(f)$ dopadne negatívne a ešte nie sme na konci súboru. Potom je $\omega(f) \wedge z=0 \wedge F=2 \wedge F \leqslant dl(f) \wedge a=f_1$ pravdivé tvrdenie, a podľa pravidla vetvenia musíme dokázať

$\{\omega(f) \wedge z=0 \wedge F=2 \wedge F \leqslant dl(f) \wedge a=f_1\}$

```
b←Read(f);
while not EOF(f)
    begin
    c←Read(f);
    if V(a,b,c)
        then z←z+1;
    a←b;
    b←c;
    end
```

$\{\psi(f,z)\}$

a) induktívny výraz

$\{\omega(f) \wedge z=0 \wedge F=2 \wedge F \leqslant dl(f) \wedge a=f_1\}$

$b \leftarrow Read(f)$

$\{\omega(f) \wedge z=0 \wedge F=3 \wedge F \leqslant dl(f)+1 \wedge a=f_2 \wedge b=f_1\}$

je pravdivý z dôvodov uvedených v bode 1.

b) teraz dokážeme platnosť formuly

$$(\omega(f) \wedge z=0 \wedge F=3 \wedge F \leqslant dl(f)+1 \wedge a=f_2 \wedge b=f_1) \Rightarrow I(f,z)$$

Predikát $\omega(f) \wedge a=f_1 \wedge b=f_2$ je triviálne pravdivý Z predpokladu $F=3 \wedge F \leqslant dl(f)+1$ vyplýva $3 \leqslant F \leqslant dl(f)+1$ a z predpokladu $F=3$ vyplýva, že množina M_F je prázdna.

c) z bodov a) a b) a použitím pravidla konsekvencie dostávame

$\{\omega(f) \wedge z=0 \wedge F=2 \wedge F \leqslant dl(f) \wedge a=f_1\}$

$b \leftarrow Read(f)$

$I(f,z)$

d) dokazované tvrdenie 11. je dôsledkom bodov 9. a 11 c) a pravidla sekvencie

12. Predpokladajme teraz, že test $F>dl(f)$ dopadne pozitívne, a teda už sme na konci súboru. Potom treba podľa pravidla vetvenia dokázať implikáciu

$$(\omega(f) \wedge z=0 \wedge F=2 \wedge F \leqslant dl(f)+1 \wedge a=f_1 \wedge F>dl(f)) \Rightarrow \psi(f,z)$$

Platnosť formuly $\omega(f) \wedge F>dl(f)$ je zrejmá. Nech množina K označuje to isté čo v bode 9. Potom musí platiť, že K je prázdna množina.

Nech teda $(\exists i)(i \in K)$. To ale z definície množiny K znamená, že $f_{i-2} \in f \wedge f_{i-1} \in f \wedge f_i \in f$. Keďže platí $\omega(f)$, potom $i \geq 3$. Čiže nutne $f_3 \in f$.

Avšak formula $F>dl(f) \wedge F=2$ implikuje platnosť formuly $dl(f)<2$, a tom prípade $f_2 \notin f$, a teda $f_3 \notin f$, čo je spor.

13. Teraz už môžeme prehlásiť dokazovaný príkaz vetvenia za pravdivý.

$\{\omega(f) \wedge z=0 \wedge F=2 \wedge F \leqslant dl(f)+1 \wedge a=f_1\}$

```
if not EOF(f)
    then begin
      b ← Read(f) ;
```

```
                while not EOF(f)
                        begin
                        c ← Read(f) ;
                        if V(a,b,c)
                                then z ← z+1 ;
                        a ← b ;
                        b ← c ;
                        end
                end
{ψ(f,z)}
```

Dôkaz vyplýva z bodov 11., 12. a pravidla vetvenia.

Predpokladajme teraz, že program sa nachádza v bode $A1$, v ktorom platí predikát

$$\omega(f) \wedge z=0 \wedge F=1 \quad .$$

Potom program vyhodnotí test $F>dl(f)$. Ak platí, program skončí, ak nie, pokračuje príslušnou vetvou. Chceme teda dokázať nasledovný výraz

```
ω(f) ∧ z=0 ∧ F=1
        if not EOF(f)
                then begin
                a ← Read(f) ;
                if not EOF(f)
                        then begin
                        b ← Read(f) ;
                        while not EOF(f)
                                begin
                                c ← Read(f) ;
                                if V(a,b,c)
                                        then z ← z+1 ;
                                a ← b ;
                                b ← c ;
                                end
                        end
                end
{ψ(f,z)}
```

Dôkaz vykonáme podobne ako dôkaz bodu 13.

14. Predpokladajme, že test $F>dl(f)$ dopadne negatívne a ešte nie sme na konci súboru. Potom je $\omega(f) \wedge z=0 \wedge F=1 \wedge F \leqslant dl(f)$ pravdivé tvrdenie, a podľa pravidla vetvenia musíme dokázať

$\omega(f) \wedge z=0 \wedge F=1 \wedge F \leqslant dl(f)$

a ← Read(f) ;

` if not EOF(f)

then begin

b ← Read(f) ;

while not EOF(f)

begin

c ← Read(f) ;

if V(a, b, c)

then z ← z + 1 ;

a ← b ;

b ← c ;

end

end

$\{\psi(f, z)\}$

Pravdivosť induktívneho výrazu

$\omega(f) \wedge z=0 \wedge F=1 \wedge F \leqslant dl(f)$

a ← Read(f) ;

$\omega(f) \wedge z=0 \wedge F=2 \wedge F \leqslant dl(f)+1 \wedge a=f_1$

je zrejmá z definície príkazu $a \leftarrow Read(f)$ a z postupu použitého v bode 1. Z neho a z bodu 13. sa dá pravidlom sekvencie dokázať čiastočná korektnosť dokazovanej vetvy príkazu.

15. Predpokladajme teraz, že test $F>dl(f)$ dopadne pozitívne, a teda už sme na konci súboru. Potom treba podľa pravidla vetvenia dokázať implikáciu

$$(\omega(f) \wedge z=0 \wedge F=1 \wedge F \leqslant dl(f)+1 \wedge F>dl(f)) \Rightarrow \psi(f, z)$$

Platnosť formuly $\omega(f) \wedge F>dl(f)$ je zrejmá. Nech množina K označuje to isté čo v bode 9. Potom musí platiť, že K je prázdna množina.

Nech teda $(\exists i)(i \in K)$. To ale z definície množiny K znamená, že $f_{i-2} \in f \wedge f_{i-1} \in f \wedge f_i \in f$.

Keďže platí $\omega(f)$, potom $i \geq 3$. Čiže nutne $f_3 \in f$.
Avšak formula $F > dl(f) \wedge F = 1$ implikuje platnosť formuly $dl(f) < 1$, a tom prípade $f_1 \notin f$, teda aj $f_3 \notin f$, čo je spor.

Dôkaz čiastočnej správnosti programu

Na dôkaz čiastočnej korektnosti potrebujeme dokázať už len niekoľko tvrdení.

Nech sa teda program nachádza na začiatku, v bode A, a nech v bode A platí vstupná podmienka špecifikácie.

16. Platnosť induktívneho výrazu

$\{\omega(f)\}$

$z \leftarrow 0$

$\{\omega(f) \wedge z = 0\}$

vyplýva z axiomy priradenia a pravidla konsekvencie.

17. Z rovnakých dôvodov platí aj

$\{\omega(f) \wedge z = 0\}$

$Reset(f)$;

$\omega(f) \wedge z = 0 \wedge F = 1$

18. Z bodov 16. a 17. dostávame použitím pravidla sekvencie

$\{\omega(f)\}$

$z \leftarrow 0$;

$Reset(f)$;

$\omega(f) \wedge z = 0 \wedge F = 1$

Teraz už môžeme pomocou pravidla sekvencie dokázať čiastočnú korektnosť celého programu.

$\{\omega(f)\}$

$z \leftarrow 0$;
$Reset(f)$;
if not $EOF(f)$
then begin ;
$a \leftarrow Read(f)$;

```
`          if not EOF(f)
               then begin ;
               b←Read(f) ;
               while not EOF(f)
                    begin
                    c←Read(f) ;
                    if V(a,b,c)
                         then z←z+1 ;
                    a←b ;
                    b←c ;
                    end
               end
          end
{ψ(f,z)}
```

Podarilo sa dokázať, že program P je čiastočne korektný vzhľadom na špecifikáciu.

V ďalšej časti budeme dokazovať, že program skončí. Ak sa nám túto skutočnosť podarí dokázať, bude to spolu s už dokázanou vlastnosťou čiastočnej korektnosti znamenať, že program P je totálne korektný vzhľadom na vstupnú podmienku $\omega(f)$ a výstupnú podmienku $\psi(f,z)$.

Dôkaz skončenia prevedieme Floydovou metódou dobre fundovaných množín.

Dôkaz totálnej správnosti

V časti „Návrh programu“ som popísal vývojový diagram programu P, z ktorého budem teraz vychádzať. V programe P sa nachádzajú tri body označené A, B, C. Bod A nazveme vstupný bod programu P, bod C výstupný bod programu P. Bodu A priradíme vstupný predikát špecifikácie $\omega(\bar{x})$, bodu B – deliacemu bodu programu P – priradíme invariant $I(f,z)$.

$$q_B(\bar{x},\bar{y}) = I(f,z)$$

Dôkaz, že $I(f,z)$ je dobrá induktívna podmienka, bol už urobený – pre každý prípustný vstup ak sa program dostane do bodu B, potom je invariant pravdivý.

Ako univerzum zoberieme množinu prirodzených čísel a štandardné usporiadanie

$$W = \mathbb{N}$$

a deliacemu bodu B priradíme funkciu $u_B(\bar{x},\bar{y})$

$$u_B(\bar{x},\bar{y}) = dl(f) - F + 1 \ .$$

Dokážeme, že $u_B(\bar{x},\bar{y})$ je dobrá funkcia. Musí platiť formula T_3 (strana 13)

$$\forall \bar{x} \forall \bar{y} [q_B(\bar{x},\bar{y}) \Rightarrow u_i(\bar{x},\bar{y}) \in W] \quad (T_3) \ .$$

Po dosadení dostaneme formulu

$$\forall \bar{x} \forall \bar{y} [I(f,z) \Rightarrow dl(f) - F + 1 \in \mathbb{N}] \quad .$$

Z platnosti invariantu v bode B vyplýva platnosť $F \leqslant dl(f) + 1$, čo je ekvivalentné formule $O \leqslant dl(f) - F + 1$, a z toho vyplýva $dl(f) - F + 1 \in \mathbb{N}$. Teda funkcia $u_B(\bar{x},\bar{y}) = dl(f) - F + 1$ je dobrá funkcia.

Zostáva overiť podmienku T_4 .

Potrebujeme zistiť, ako sa hodnota funkcie $u_B(\bar{x},\bar{y}) = dl(f) - F + 1$ počas jedného prechodu cyklom zmení. Keďže dl(f) je konštanta, zaujíma nás zmena hodnoty F . V cylke sa nachádza jedinný príkaz, ktorý mení hodnotu F , a je to príkaz $c \leftarrow Read(f)$, počas ktorého sa hodnota F zvýši o jednotku.

Preto po dosadení formula T_4 pre deliaci bod B programu P vyzerá takto:
$(\omega(f) \wedge 3 \leqslant F \leqslant dl(f) \wedge a = f_{F-2} \wedge b = f_{F-1} \wedge z = |M_F|) \Rightarrow [dl(f) + 1 - F > dl(f) + 1 - (F+1)]$.

Formula $[dl(f) + 1 - F > dl(f) + 1 - (F+1)]$ je pravdivá, lebo sa dá ekvivalentne upraviť na formulu 1 > 0.

To znamená, že program P končí a keďže P je aj čiastočne správny, podarilo sa nám dokázať hlavný cieľ: program P je totálne korektný vzhľadom na špecifikáciu.

Záver

V tejto práci bol podaný prehľad o metóde, ktorou možno verifikovať programy. Samozrejme, existujú aj iné metódy používané na verifikáciu, ako je napríklad Floydova metóda induktívnych podmienok alebo metóda intermedientov [3]. Tieto metódy nie sú v práci na inom mieste spomínané.

Ďalej bol vykonaný dôkaz správnosti programu, ktorý ukázal praktickú aplikovateľnosť uvedených teoretických metód dokazovania: Hoareovej a metódy dobre fundovaných množín.

Čitateľovi možno prišla na um otázka, či sa programy nedajú verifikovať automaticky. Inak povedané, zaujíma nás, či existuje algoritmus, ktorý pre ľubovoľnú trojicu $(\omega, \psi, \mathrm{P})$ povie, či je program P totálne správny vzhľadom na ω, ψ. Odpoveď je negatívna, taký algoritmus neexistuje [2]. Avšak existujú automatické verifikátory pre niektoré triedy programov, inak povedané, pre vhodne zvolené obmedzenia na trojicu $(\omega, \psi, \mathrm{P})$ sa dajú nájsť verifikačné algoritmy. Pokiaľ nechceme trojice $(\omega, \psi, \mathrm{P})$ obmedzovať, musíme sa uspokojiť s interaktívnymi verifikátormi, ktoré na úspešné verifikovanie potrebujú pomoc človeka, ktorý musí uhádnuť vnútorné špecifikácie v konkrétnych bodoch programu.

Keďže je verifikácia algoritmicky neriešiteľná úloha, o to viac je potrebné, aby sa v našej spoločnosti vyskytovali ľudia, ktorí verifikovať vedia. Táto práca by mala k tomu pomôcť.

Literatúra

[1] http://en.wikipedia.org/wiki/C._A._R._Hoare

[2] Manna, Z.: Matematická teorie programů. SNTL, Praha, 1981, s. 157- 202.

[3] Prívara, I.: Základy teórie programovania. S 47 – 53.