Verifikacija programa

— *Pregled poglavlja ,,Verification of Programs" knjige* Mathematical Theory of Computation*—

Milena Vujošević-Janičić
Jelena Tomašević

*Zohar Manna: *Mathematical Theory of Computation*, McGraw-Hill, 1974.

Sadržaj

1 Uvod

Pretpostavimo da je dat program sa opisom njegovog ponašanja, odnosno sa jasno definisanim *izlaznim predikatom*, koji opisuje relacije između promenljivih programa koje moraju da budu zadovoljene na kraju rada programa. Nekada je takođe dat i *ulazni predikat* koji definiše ulazne uslove koji moraju da budu zadovoljeni kako bi izvršavanje programa imalo smisla. Zadatak je dokazati da je program korektan u odnosu na dati ulazni i izlazni predikat, tj. dokazati *da za svako izvršavanje programa za koje ulazne vrednosti zadovoljavaju ulazni predikat, može da se garantuje da će nakon izvršavanja programa biti zadovoljen izlazni predikat.*

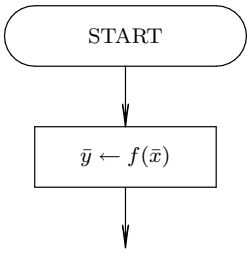
2 Dijagramski programi

Uočimo tri vrste promenljivih, grupisane u vektore:

1. *Ulazni vektor* $\bar{x} = (x_1, x_2, ..., x_n)$ se sastoji od datih ulaznih vrednosti i ne menja se za vreme izvršavanja programa. Ulazni vektor je definisan nad ulaznim domenom $D_{\bar{x}} = D_{x_1} \times D_{x_2} \times ... \times D_{x_n}$.
2. *Programski vektor* $\bar{y} = (y_1, y_2, ..., y_n)$ se koristi za privremeno skladištenje rezultata prilikom izvršavanja programa. Programski vektor je definisan nad programskim domenom $D_{\bar{y}} = D_{y_1} \times D_{y_2} \times ... \times D_{y_n}$.
3. *Izlazni vektor* $\bar{z} = (z_1, z_2, ..., z_n)$ sadrži izlazne vrednosti kada se izvršavanje programa završi. Izlazni vektor je definisan nad izlaznim domenom $D_{\bar{z}} = D_{z_1} \times D_{z_2} \times ... \times D_{z_n}$.

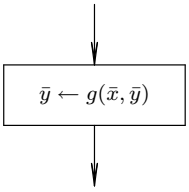
Četiri osnovne naredbe dijagramskog programa su

1. START naredba



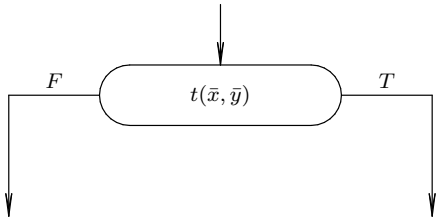
   gde je $f(\bar{x}) : D_{\bar{x}} \longrightarrow D_{\bar{y}}$ funkcija definisana za sve vrednosti domena $D_{\bar{x}}$.

2. naredba DODELE



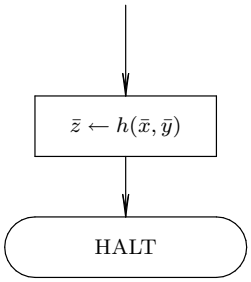
gde je $g(\bar{x}, \bar{y}) : D_{\bar{x}} \times D_{\bar{y}} \longrightarrow D_{\bar{y}}$ funkcija definisana za sve vrednosti iz $D_{\bar{x}} \times D_{\bar{y}}$.

3. naredba USLOVA



gde je $t(\bar{x}, \bar{y})$ predikat definisan za sve vrednosti iz $D_{\bar{x}} \times D_{\bar{y}}$.

4. naredba ZAUSTAVLJANJA (ili HALT naredba)



gde je $h(\bar{x}, \bar{y}) : D_{\bar{x}} \times D_{\bar{y}} \longrightarrow D_{\bar{z}}$ funkcija definisana za sve vrednosti iz $D_{\bar{x}} \times D_{\bar{y}}$.

Definicija 1 *Dijagramski program je proizvoljan dijagram konstruisan od osnovne četiri naredbe tako da važi:*

- *postoji tačno jedna START naredba;*
- *svaka naredba DODELE ili USLOVA nalazi se na putu od naredbe START do neke naredbe ZAUSTAVLJANJA.*

Program može da se izvrši ako je data vrednost $\bar{\xi} \in D_{\bar{x}}$ za ulazni vektor $\bar{x}$.

1. Izvršavanje uvek počinje od START naredbe tako što se inicijalizuje vrednost $\bar{y}$ sa $f(\bar{\xi})$.

2. Izvršavanje sledi granu do sledeće naredbe

   i to tako što

   - kad god se dođe do naredbe DODELE, vredost $\bar{y}$ se zamenjuje sa $g(\bar{x}, \bar{y})$,
   - kad god se dođe do naredbe USLOVA izvršavanje prati T ili F granu, u zavisnosti od toga da li je vrednost $t(\bar{x}, \bar{y})$ tačno ili netačno pri čemu vrednost $\bar{y}$ ostaje nepromenjena.

3. Ukoliko se prilikom izvršavanja dijagramskog programa naiđe na naredbu ZAUSTAVLJANJA tada se izlaznom vektoru $\bar{z}$ dodeljuje tekuća vrednost $\bar{\zeta}$ funkcije $h(\bar{x}, \bar{y})$ i tada je $P(\bar{\xi})$ definisano i $P(\bar{\xi}) = \bar{\zeta}$, a u suprotnom — ako se izvršavanje programa nikada ne završava — kažemo da je $P(\bar{\xi})$ nedefinisano.

2.1 Verifikacija programa

Verifikacija dijagramskog programa zavisi od ulaznog i izlaznog predikata.

Ulazni predikat $\varphi(\bar{x})$ definisan na domenu $D_{\bar{x}}$ opisuje elemente domena $D_{\bar{x}}$ koji mogu da budu korišćeni kao ulazne vrednosti programa. Nas, dakle, interesuje ponašanje našeg programa samo za one elemente domena $D_{\bar{x}}$ za koje je $\varphi(\bar{x})$ tačno.

Izlazni predikat $\psi(\bar{x}, \bar{z})$ definisan na $D_{\bar{x}} \times D_{\bar{z}}$ opisuje odnos koji mora da bude zadovoljen između ulaznih i izlaznih promenljivih nakon izvršavanja programa.

Definicija 2 *Program P se* zaustavlja u odnosu na ulazni predikat φ *ako se njegovo izvršavanje zaustavlja za svaku ulaznu vrednost $\bar{\xi}$ takvu da je $\varphi(\bar{\xi})$ tačno.*

Definicija 3 *Program P je* parcijalno korektan *u odnosu na φ i ψ ako za svaku ulaznu vrednost $\bar{\xi}$ takvu da je $\varphi(\bar{\xi})$ tačno i izvršavanje programa se zaustavlja, važi da je $\psi(\bar{\xi}, P(\bar{\xi}))$ tačno.*

Definicija 4 *Program P je* tatalno korektan *u odnosu na φ i ψ ako za svaku ulaznu vrednost $\bar{\xi}$ takvu da je $\varphi(\bar{\xi})$ tačno, važi da se izrvšavanje programa zaustavlja i da je $\psi(\bar{\xi}, P(\bar{\xi}))$ tačno.*

Primetimo da je razlika između parcijalne i totalne korektnosti u tome što se za totalnu korektnost garantuje zaustavljanje programa ukoliko je $\varphi(\bar{\xi})$ tačno, dok se za parcijalnu korektnost to ne garantuje.

Definicija 5 *Verifikacija programa P za dati ulazni predikat $\varphi(\bar{x})$ i izlazni predikat $\psi(\bar{x}, \bar{z}))$ je dokazivanje totalne korektnosti programa u ondosu na φ i ψ.*

Verifikaciju programa najčešće je lakše dokazati u dva odvojena koraka:

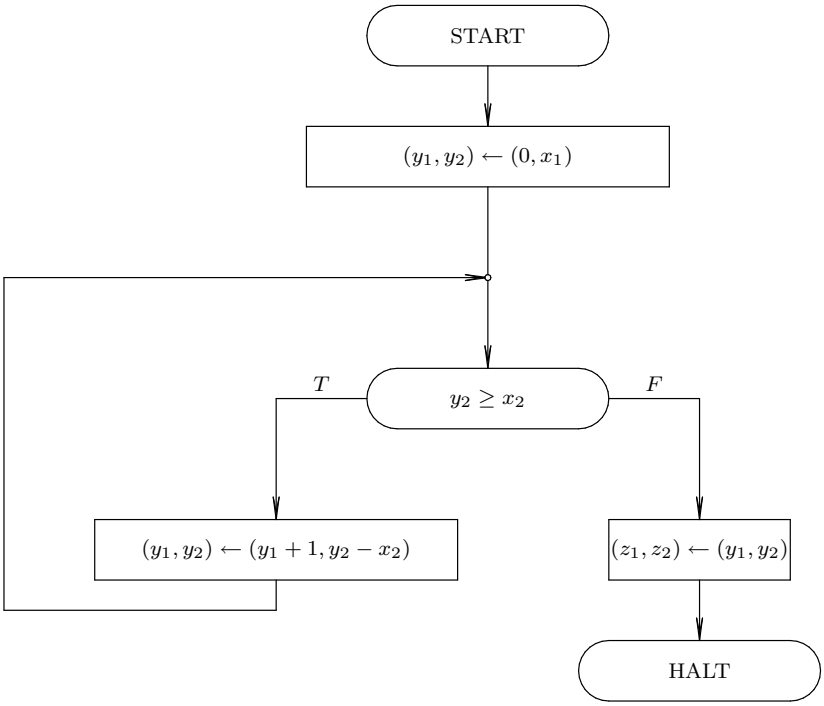
1. dokazivanje parcijalne korektnosti programa u odnosu na φ i ψ;
2. dokazivanje zaustavljanja programa u odnosu na φ.

Primer 1 (Celobrojno deljenje — dokaz totalne korektnosti) *Dokazati totalnu korektnost programa za celobrojno deljenje dva broja.*

Dijagramski program na slici 1 izračunava celobrojni količnik z_1 i ostatak z_2 prilikom deljenja promenljive x_1 sa promenljivom x_2, pri uslovu da je $x_1 \geq 0$ i $x_2 > 0$. U ovom primeru je $\bar{x} = (x_1, x_2)$, $\bar{y} = (y_1, y_2)$, $\bar{z} = (z_1, z_2)$ i $D_{\bar{x}} = D_{\bar{y}} = D_{\bar{z}} = \{\text{svi parovi celih brojeva}\}$.

Najpre ćemo pokazati da je program parcijalno korektan u odnosu na ulazni predikat

$$\varphi(x_1, x_2) : x_1 \geq 0 \wedge x_2 \geq 0$$



Slika 1: Dijagramski program za izračunavanje celobrojnog deljenja

(što pokazuje da smo zainteresovani samo za izvršavanje programa kada su obe promenljive nenegativne) i izlazni predikat

$$\psi(x_1, x_2, z_1, z_2) : x_1 = z_1 x_2 + z_2 \wedge 0 \leq z_2 < x_2$$

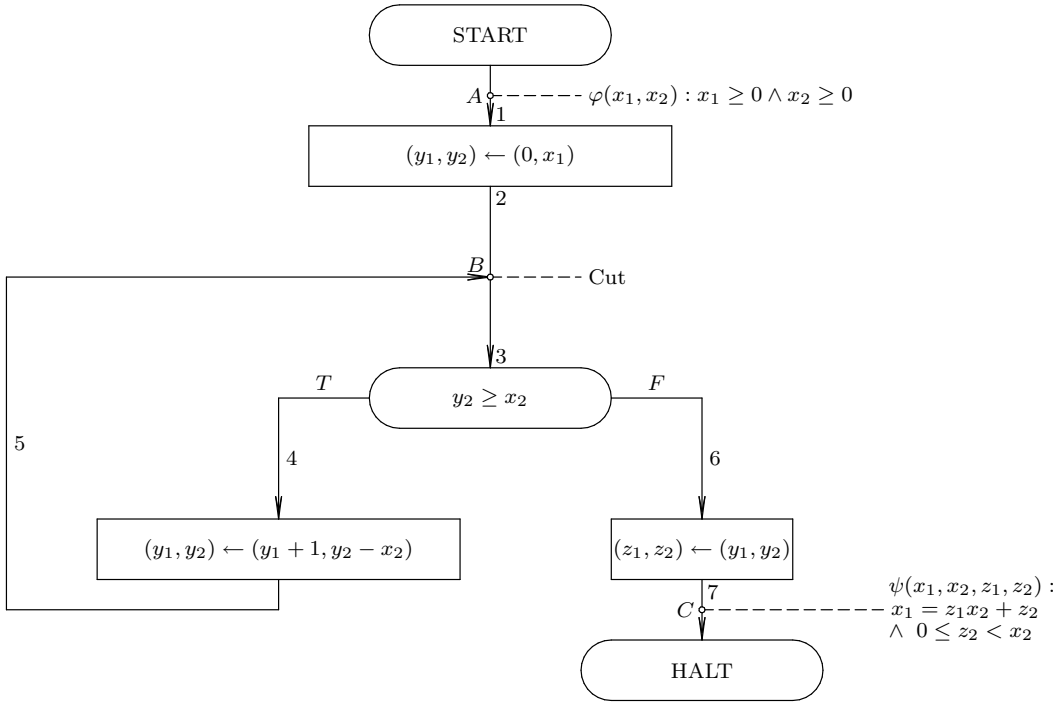
(koji zapravo i predstavlja definiciju celobrojnog deljenja).

Zatim ćemo pokazati da se program zaustavlja u odnosu na

$$\varphi'(x_1, x_2) : x_1 \geq 0 \wedge x_2 > 0.$$

Ako dokažemo da je program parcijalno korektan u odnosu na φ i ψ i da se zaustavlja u odnosu na φ', sledi da je program totalno korektan u odnosu na φ' i ψ. Suštinska razlika između φ i φ' je u isključivanju slučaja $x_2 = 0$ jer se tada program ne zaustavlja.

Da bismo dokazali parcijalnu korektnost programa celobrojnog deljenja u odnosu na φ i ψ pridružićemo ulazni predikat φ tački A, a izlazni predikat ψ tački C dijagrama (pogledati sliku 2). Glavni problem u verifikaciji programa je obraditi petlje. Petlja ovog programa postaje podesna za rukovanje presecanjem programa u tački B čime se tok programa deli na tri putanje: prvu označimo sa α (POČETAK) (od tačke A do tačke B(strelice 1 i 2 na slici 2)), drugu sa β (PETLJA) (od tačke B do tačke B (strelice 3,4,5)) i treću sa γ (KRAJ) (od tačke B do C (strelice 3,6,7)). Sva moguća izvršenja programa moraju prvo proći putanju α, zatim nula ili više puta putanju β i na kraju završiti



Slika 2: Dijagramski program za izračunavanje celog dela broja x

sa putanjom γ, što znači da su sva izvršenja programa ,,pokrivena“ sa ove tri putanje.

U cilju dokazivanja parcijalne korektnosti programa, odredimo predikat $p(x_1, x_2, y_1, y_2)$ koji će opisivati odnos između programskih promenljivih u tački B. Podesan predikat za ovu svrhu je

$$x_1 = y_1 x_2 + y_2 \wedge y_2 \geq 0.$$

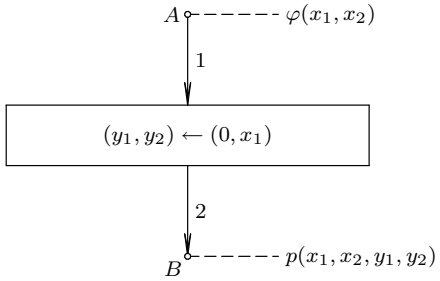
Na ovaj način program je ,,pokriven“ sa tri putanje od kojih svaka počinje i završava se predikatom. Parcijalna korektnost se dokazuje proverom svake od tri petlje i dokazom da ako je početni predikat za tu petlju tačan za neke vrednosti $\bar{x}$ i $\bar{y}$, onda je, nakon prolaska kroz tu putanju, završni predikat takođe tačan ali za nove vrednosti $\bar{x}$ i $\bar{y}$.

U programu celobrojnog deljenja, proverom putanje α može se utvrditi da je predikat $p(x_1, x_2, y_1, y_2)$ tačan pri prvom ulasku u petlju programa (pretpostavlja se da je ulazni predikat zadovoljen). Proverom putanje β može se pokazati da ako je $p(x_1, x_2, y_1, y_2)$ tačno pri ulasku u petlju prvi put, biće tačno i drugi put, ako je tačno drugi put biće tačno i treći put i tako dalje. Proverom putanje γ može se utvrditi da ako je petlja napuštena tako da je predikat $p(x_1, x_2, y_1, y_2)$ tačan, onda će i izlazni predikat biti tačan. Predikat $p(x_1, x_2, y_1, y_2)$ se još naziva i *invarijanta petlje*.

U cilju kompletiranja dokaza parcijalne korektnosti programa celobrojnog

deljenja, moramo izvršiti gore pomenute provere putanja α, β i γ. Ove provere se sastoje iz dva dela: prvo se konstruišu uslovi provere u terminima zadatih predikata, a zatim se dokazuju. Konstruisanje uslova provere putanje obično se vrši vraćanjem unazad kroz putanju razmatrajući svaki iskaz redom.

1. *Putanja α*



Postavlja se pitanje šta je to što mora biti tačno u tački A tako da posle izvršetka iskaza $(y_1, y_2) \leftarrow (0, x_1)$, budemo sigurni da će predikat $p(x_1, x_2, y_1, y_2)$ biti tačan. Odgovor je $p(x_1, x_2, 0, x_1)$, koji je formiran zamenom svake pojave y_1 u predikatu sa nulom a y_2 sa x_1. Dakle, uslov provere za putanju α je

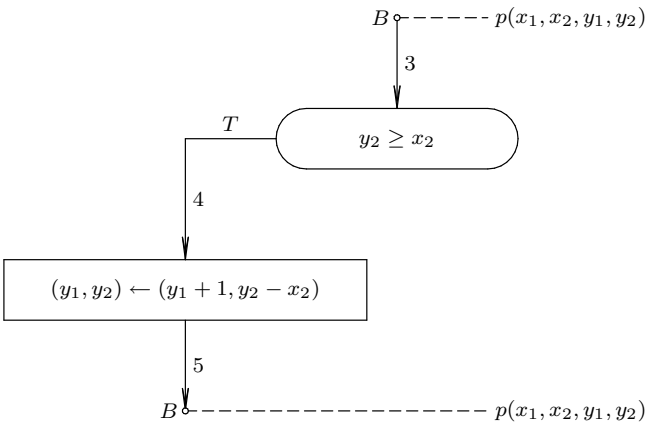
$$\varphi(x_1, x_2) \Rightarrow p(x_1, x_2, 0, x_1)$$

a to se svodi na

$$[x_1 \geq 0 \wedge x_2 \geq 0] \Rightarrow [x_1 = 0 \cdot x_2 + x_1 \wedge x_1 \geq 0]$$

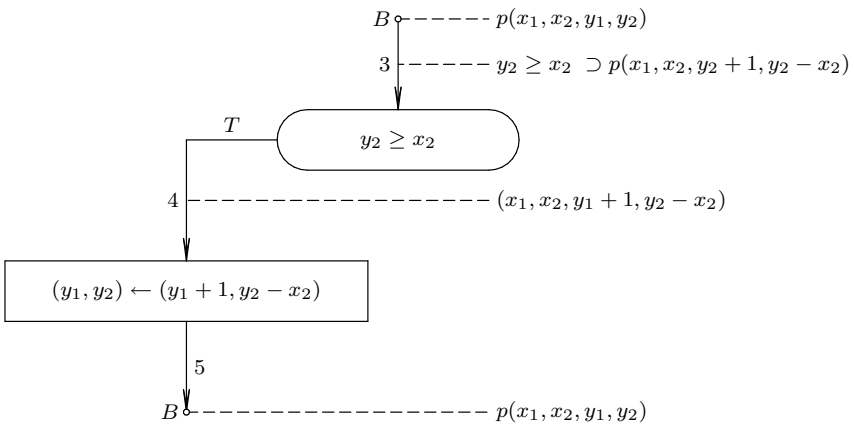
2. *Putanja β*

Da bi se konstruisao predikat provere za putanju β, uslov naredbe USLOVA mora biti zadovoljen.



Predikat $p(x_1, x_2, y_1, y_2)$ se izvodi prolaskom unazad i zamenom $(y_1, y_2) \leftarrow (y_1 + 1, y_2 - x_2)$ čime se dobija $p(x_1, x_2, y_1 + 1, y_2 - x_2)$. Iako naredba

USLOVA ne menja vrednosti nijedne programske promenljive, ona daje dodatne korisne informacije zato što je nakon testa sigurno da važi $y_2 \geq x_2$. Opet se postavlja isto pitanje kao i za putanju α: šta je to što treba da bude tačno na strelici 3 tako da kad uslov naredbe USLOVA bude tačan (tj. kada je $y_2 \geq x_2$ tačno), da i predikat $p(x_1, x_2, y_1 + 1, y_2 - x_2)$ bude tačan na strelici 4? U ovom slučaju odgovor je jasan: to je $y_2 \geq x_2 \Rightarrow p(x_1, x_2, y_1 + 1, y_2 - x_2)$. Kompletna analiza putanje β je predstavljena slikom:



U ovom slučaju uslov provere koji treba da se dokaže je

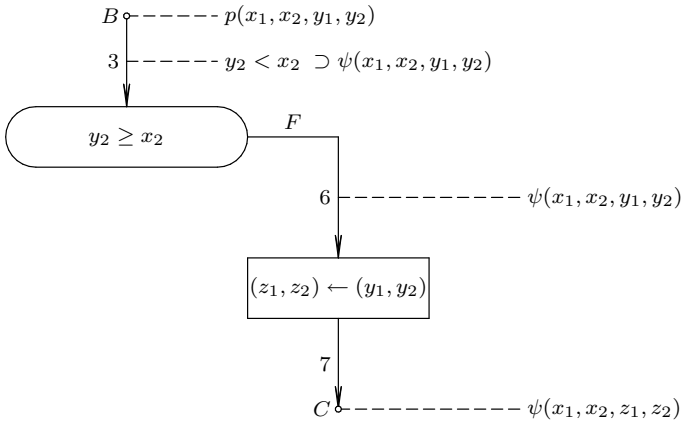
$$p(x_1, x_2, y_1, y_2) \Rightarrow [y_2 \geq x_2 \Rightarrow p(x_1, x_2, y_1 + 1, y_2 - x_2)]$$

što je ekvivalentno sa

$$p(x_1, x_2, y_1, y_2) \wedge y_2 \geq x_2 \Rightarrow p(x_1, x_2, y_1 + 1, y_2 - x_2)$$

3. *Putanja γ*

Rezultat analize putanje γ dat je na sledećoj slici:



Uslov provere koji treba dokazati je

$$p(x_1, x_2, y_1, y_2) \Rightarrow [y_2 < x_2 \Rightarrow \psi(x_1, x_2, y_1, y_2)]$$

što je ekvivalentno sa

$$p(x_1, x_2, y_1, y_2) \wedge y_2 < x_2 \Rightarrow \psi(x_1, x_2, y_1, y_2).$$

Dakle, formirali smo tri uslova provere:

1. $\varphi(x_1, x_2) \Rightarrow p(x_1, x_2, 0, x_1)$ (α)
2. $p(x_1, x_2, y_1, y_2) \wedge y_2 \geq x_2 \Rightarrow p(x_1, x_2, y_1 + 1, y_2 - x_2)$ (β)
3. $p(x_1, x_2, y_1, y_2) \wedge y_2 < x_2 \Rightarrow \psi(x_1, x_2, y_1, y_2)$ (γ)

pri čemu je:
$\varphi(x_1, x_2) : x_1 \geq 0 \wedge x_2 \geq 0$
$p(x_1, x_2, y_1, y_2) : x_1 = y_1 x_2 + y_2 \wedge y_2 \geq 0$
$\psi(x_1, x_2, z_1, z_2) : x_1 = z_1 x_2 + z_2 \wedge 0 \leq z_2 < x_2$

Kako su za sve cele brojeve x_1, x_2, y_1, y_2 sva tri uslova provere ispunjena, program je parcijalno korektan u odnosu na φ i ψ.

Za sada smo dokazali da kada se izvršavanje programa zaustavi onda je izlazni predikat tačan. No, nismo pokazali da će se izvršavanje programa ikada zaustaviti. Dakle, da bismo verifikovali program celobrojnog deljenja, neophodno je dokazati i njegovo zaustavljanje.

Zaustavljanje ćemo dokazati u odnosu na

$$\varphi'(x_1, x_2) : x_1 \geq 0 \wedge x_2 > 0.$$

Dokažimo najpre da predikat

$$q(x_1, x_2, y_1, y_2) : y_2 \geq 0 \wedge x_2 > 0$$

ima osobinu da kad god se dospe u tačku B za vreme izvršavanja programa, onda je $q(x_1, x_2, y_1, y_2)$ tačno za tekuće vrednosti promenljivih. Da bismo ovo pokazali, moramo prvo da dokažemo sledeća dva verifikaciona uslova:

$$\varphi'(x_1, x_2) \Rightarrow q(x_1, x_2, 0, x_1)$$

$$q(x_1, x_2, y_1, y_2) \wedge y_2 \geq x_2 \Rightarrow q(x_1, x_2, y_1 + 1, y_2 - x_2).$$

Ova dva predikata svode se na uslove

$$(x_1 \geq 0 \wedge x_2 > 0) \Rightarrow (x_1 \geq 0 \wedge x_2 > 0)$$

$$(y_2 \geq 0 \wedge x_2 > 0 \wedge y_2 \geq x_2) \Rightarrow (y_2 - x_2 \wedge x_2 > 0)$$

koji su očigledno tačni.

Primetimo da pošto uslov $x_2 > 0$ uvek važi u tački B, svaki put kad se prođe kroz petlju (od tačke B do tačke B), *vrednost promenljive y_2 opada*. Takođe, važi da je $y_2 \geq 0$ kada se god dođe u tačku B. Kako u skupu prirodnih brojeva ne postoji opadajući niz elemenata takav da je $a_0 > a_1 > a_2 > ...$, ne može se beskonačno mnogo puta proći kroz petlju. Drugim rečima, program se zaustavlja.

Koristeći ovaj prilaz, možemo definisati opštu tehniku za izvođenje dokaza korektnosti programa.

2.2 Parcijalna korektnost

Dokaz parcijalne korektnosti se sastoji iz tri koraka:

1. Određivanje presečnih tačaka programa kojim se izdvajaju sve moguće putanje izvršenja programa.
2. Pridruživanje tim tačkama odgovarajućih induktivnih tvrđenja.
3. Za svaku moguću putanju između dve tačke u programu konstruisanje odgovarajuće relacije.

Teorema 1 (Metod induktivnih tvrđenja (Floyd)) *Ako na dati program P, ulazni predikat $\varphi(\bar{x})$ i izlazni predikat $\psi(\bar{x}, z)$ primenimo navedena tri koraka i ukoliko su sve relacije konstruisane u trećem koraku tačne, onda kažemo da je program parcijalno korektan u odnosu na ulazni predikat $\varphi(\bar{x})$ i izlazni predikat $\psi(\bar{x}, z)$.*

Pre navođenja primera dokaza parcijalne korektnosti programa, uvedimo nekoliko novih oznaka. Neka je α oznaka za putanju od i-te do j-te presečne tačke. Sa $R_\alpha(\bar{x}, \bar{y})$ označavaćemo predikat koji iskazuje uslov koji treba biti ispunjen da bi se prešla ova putanja, a sa $r_\alpha(\bar{x}, \bar{y})$ predikat koji opisuje transformaciju promenljive $\bar{y}$ pri prelasku putanje α. Jednostavan metod za određivanje vrednosti za R_α i r_α je *tehnika zamenjivanja unazad*: inicijalno, $R_\alpha(\bar{x}, \bar{y})$ se postavlja da bude T, a $r_\alpha(\bar{x}, \bar{y})$ da bude $\bar{y}$ i oba su pridružena presečnoj tački j. U svakom koraku se te stare vrednosti za $R_\alpha(\bar{x}, \bar{y})$ i $r_\alpha(\bar{x}, \bar{y})$ koriste kako bi se izračunale nove. Konačne vrednosti za R i r dobijene u presečnoj tački i su željene vrednosti R_α i r_α za tu putanju.

Primer 2 (Celobrojni koren — dokaz parcijalne korektnosti) *Dokazati parcijalnu korektnost programa koji za svaki prirodan broj x izračunava celobrojni koren broja $z = \lfloor\sqrt{x}\rfloor$, odnosno najveći ceo broj k takav da je $k \leq \sqrt{x}$.*

Dijagramski program koji rešava ovaj problem dat je na slici 3.

Metod korišćen za rešavanje ovog problema baziran je na činjenici da je $1{+}3{+}5{+}\ldots{+}(2{\cdot}n{+}1) = (n{+}1)^2$ za svako $n \geq 0$. Pri tom se n izračunava u okviru promenljive y_1, neparni brojevi $2{\cdot}n{+}1$ u okviru y_3 a suma $1{+}3{+}5{+}\ldots{+}(2{\cdot}n{+}1)$ u okviru promenljive y_2.

Dokažimo parcijalnu korektnost ovog programa u odnosu na ulazni predikat $\varphi(x) : x \geq 0 \wedge x_2 > 0$ koji ćemo pridružiti tački A na slici 4 i izlazni predikat $\psi(x, z) : z^2 \leq x \leq (z+1)^2$ koji ćemo pridružiti tački C.

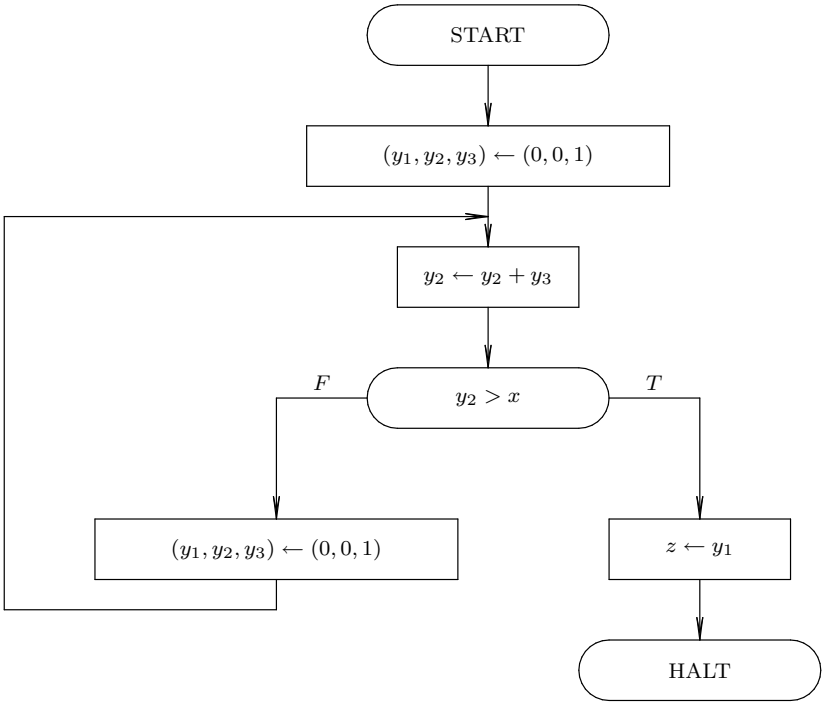
Prvi korak u ostvarenju tog cilja je određivanje presečne tačke programa B kojom će se izdvojiti sve moguće putanje izvršenja programa.

U *drugom koraku*, tački B ćemo pridružiti sledeće induktivno tvrđenje:

$$p(x, y_1, y_2, y_3) : y_1^2 \leq x \wedge y_2 = (y_1 + 1)^2 \wedge y_3 = 2 \cdot y_1 + 1$$

U okviru *trećeg koraka*, da bismo dokazali parcijalnu korektnost programa, potrebno je još da za svaku moguću putanju između dve tačke u programu konstruišemo odgovarajuće relacije.

U ovom programu možemo razlikovati sledeće tri putanje:



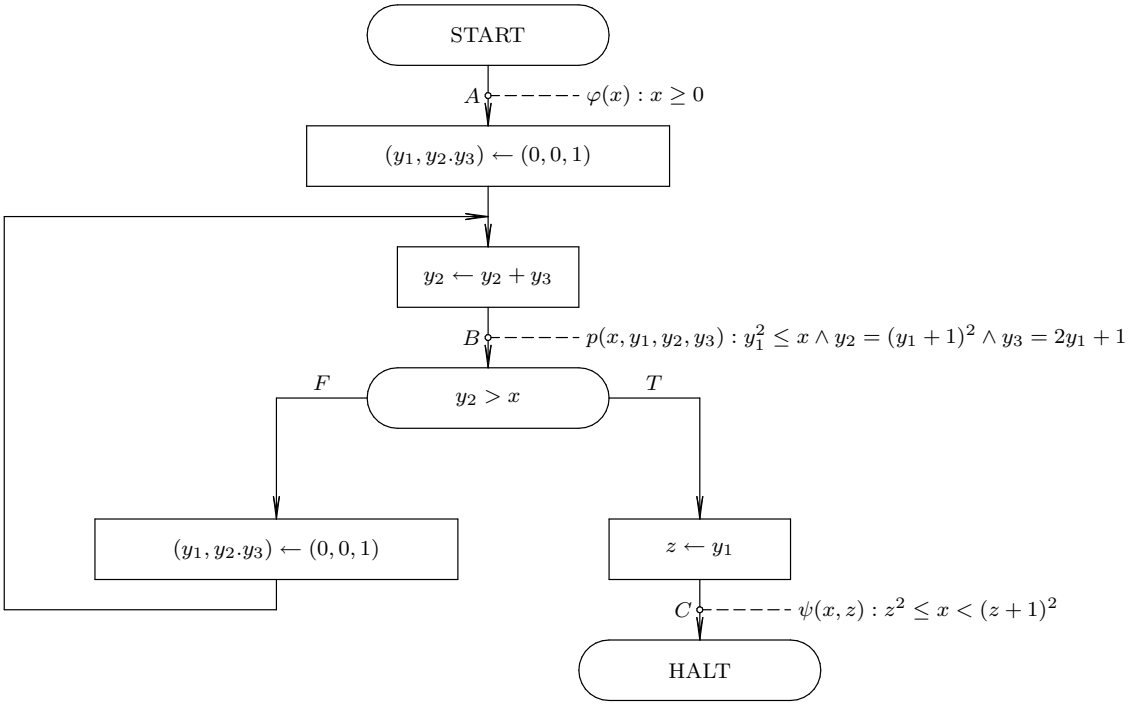
Slika 3: Dijagramski program za izračunavanje celobrojnog korena broja x

1. *Putanja* α (od tačke A do tačke B).
   Njoj odgovarajuća relacija je
   $[\varphi(x) \wedge \top] \Rightarrow p(x, 0, 1, 1)$
   koja se svodi na
   $x \geq 0 \Rightarrow [0^2 \leq x \wedge 1 = (0+1)^2 \wedge 1 = 2 * 0 + 1]$

2. *Putanja* β (od tačke B do tačke B).
   Njoj odgovarajuća relacija je
   $[p(x, y_1, y_2, y_3) \wedge y_2 \leq x] \Rightarrow p(x, y_1 + 1, y_2 + y_3 + 2, y_3 + 2)$
   koja se svodi na
   $[y_1^2 \leq x \wedge y_2 = (y_1+1)^2 \wedge y_3 = 2 * y_1 + 1 \wedge y_2 \leq x] \Rightarrow [(y_1+1)^2 \leq x \wedge y_2 + y_3 + 2 = (y_1+2)^2 \wedge y_3 + 2 = 2 * ((y_1+1)+1]$

3. *Putanja* γ (od tačke B do tačke C).
   Njoj odgovarajuća relacija je
   $[p(x, y_1, y_2, y_3) \wedge y_2 > x] \Rightarrow \psi(x, y_1)$
   koja se svodi na
   $[y_1^2 \leq x \wedge y_2 = (y_1+1)^2 \wedge y_3 = 2 * y_1 + 1 \wedge y_2 > x] \Rightarrow y_1^2 \leq x < (y_1+1)^2$

Kako su sve ove relacije tačne, na osnovu Floyd-ove metode o induktivnom tvrđenju, naš program je parcijalno korektan u odnosu na ulazni predikat $\varphi(x)$: $x \geq 0 \wedge x_2 > 0$ i izlazni predikat $\psi(x, z) : z^2 \leq x \leq (z+1)^2$.



Slika 4: Dijagramski program za izračunavanje celobrojnog korena broja x

2.3 Zaustavljanje

Pre ilustrovanja metode za dokazivanje zaustavljanja programa, uvedimo pojam *dobro-zasnovanog skupa*.

Definicija 6 *Za uređeni par $(W, \prec)$, gde je W neki neprazan skup a $\prec$ neka binarna relacija definisana na skupu W, kažemo da je* parcijalno uređen skup *ako zadovoljava sledeće uslove:*

1. Tranzitivnost: *ako je $a \prec b$ i $b \prec c$ onda je i $a \prec c$, za sve $a, b, c \in W$*
2. Antisimetričnost: *ako je $a \prec b$ onda nije $b \prec a$, za sve $a, b \in W$*
3. Nerefleksivnost: *ne važi $a \prec a$ ni za jedno $a \in W$.*

Definicija 7 *Za uređeni par $(W, \prec)$, gde je W neki neprazan skup a $\prec$ neka binarna relacija definisana na skupu W, kažemo da je* dobro zasnovan skup *ako je W parcijalno uređen i ako ne postoji beskonačan opadajući niz elemenata iz W takav da je $a_0 \succ a_1 \succ a_2 \succ ...$*

Pretpostavimo da imamo zadat dijagramski program i ulazni predikat φ. Dokaz da se program zaustavlja za ulazni predikat φ se sastoji iz tri koraka:

1. Određivanje skupa presečnih tačaka programa i svakoj takvoj tački pridruživanje tvrđenja $q_i(\bar{x}, \bar{y})$ takvog da je to *dobro tvrđenje*. To znači da za

svaku putanju α od startne tačke pa do j-te presečne tačke (bez posrednih tačaka) važi:
$\forall \bar{x}[\varphi(\bar{x}) \wedge R_\alpha(\bar{x}) \Rightarrow q_j(\bar{x}, r_\alpha(\bar{x}))]$
i da za svaku putanju α od presečne tačke i do j (bez posrednih tačaka) važi
$\forall \bar{x} \forall \bar{y}[q_i(\bar{x}, \bar{y}) \wedge R_\alpha(\bar{x}, \bar{y}) \Rightarrow q_j(\bar{x}, r_\alpha(\bar{x}, \bar{y}))]$

2. Određivanje *dobro-zasnovanog* skupa $(W, \prec)$ i svakoj presečnoj tački pridruživanje parcijalne funkcije $u_i(\bar{x}, \bar{y})$ koja preslikava $D_{\bar{x}} \times D_{\bar{y}}$ u skup W tako da je $u_i(\bar{x}, \bar{y})$ *dobra funkcija*. To znači da za svaku presečnu tačku i važi:
$\forall \bar{x} \forall \bar{y}[q_i(\bar{x}, \bar{y}) \Rightarrow u_i(\bar{x}, \bar{y}) \in W]$

3. Dokazivanje da su *uslovi zaustavljanja podržani*. To znači da za svaku putanju α od presečne tačke i do j (bez posrednih tačaka) koja je deo neke petlje važi:
$\forall \bar{x} \forall \bar{y}\{q_i(\bar{x}, \bar{y}) \wedge R_\alpha(\bar{x}, \bar{y}) \Rightarrow [u_i(\bar{x}, \bar{y}) \succ u_j(\bar{x}, r_\alpha(\bar{x}, \bar{y}))]\}$

Teorema 2 (Metod dobro zasnovanih skupova (Floyd)) *Ako na dati program P i na ulazni predikat $\varphi(\bar{x})$ primenimo navedena tri koraka i ukoliko su svi uslovi zaustavljanja konstruisani u trećem koraku tačni, onda se program zaustavlja u odnosu na ulazni predikat $\varphi(\bar{x})$.*

Primer 3 (Celobrojni koren — dokaz zaustavljanja programa) *Dokazati da se program iz primera 2 koji za svaki prirodan broj x izračunava ceo deo korena tog broja $z = \lfloor \sqrt{x} \rfloor$, zaustavlja.*

U cilju dokazivanja da se program zaustavlja u odnosu na ulazni predikat $\varphi(x) : x \geq 0 \wedge x_2 > 0$ izabraćemo dobro zasnovan skup $(N, <)$ gde je N skup prirodnih brojeva sa uobičajenim uređenjem u odnosu na relaciju $<$. Jedinu petlju u algoritmu ćemo podeliti tačkom B kojoj ćemo pridružiti tvrđenje $q(x, \bar{y}) : y_2 \leq x \wedge y_3 > 0$ i funkciju $u(x, \bar{y}) : x - y_2$.

Možemo primetiti da postoje dve putanje koje su od interesa:

1. *Putanja* α_1 (od tačke A do tačke B):
R_{α_1} je T a r_{α_1} je (0,0,1)

2. *Putanja* α_2 (od tačke B do tačke B):
R_{α_2} je $y_2 + y_3 \leq x$ a r_{α_2} je $(y_1 + 1, y_2 + y_3, y_3 + 2)$

Dokaz se sastoji iz tri koraka, za sve cele brojeve x, y_1, y_2 i y_3:

1. $q(x, \bar{y})$ je dobro tvrđenje.

   Za putanju α_1

   $\varphi(x) \Rightarrow q(x, 0, 0, 1)$

   tj.

   $x \geq 0 \Rightarrow [0 \leq x \wedge 1 > 0]$.

   Za putanju α_2

   $[q(x, y_1, y_2, y_3) \wedge y_2 + y_3 \leq x] \Rightarrow q(x, y_1 + 1, y_2 + y_3, y_3 + 2)$

   tj.

   $[y_2 \leq x \wedge y_3 > 0 \wedge y_2 + y_3 \leq x] \Rightarrow [y_2 + y_3 \leq x \wedge y_3 + 2 > 0]$.

2. $u(x, \bar{y})$ je dobra funkcija.

   $q(x, \bar{y}) \Rightarrow u(x, \bar{y}) \in N$

   tj.

   $[y_2 \leq x \wedge y_3 > 0] \Rightarrow x - y_2 \geq 0.$

3. Uslovi zaustavljanja su zadovoljeni.

   Za putanju α_2

   $[q(x, y_1 \wedge y_2 + y_3 \leq x] \Rightarrow [u(x, y_1, y_2, y_3) > u(x, y_1 + 1, y_2 + y_3, y_3 + 2)]$

   tj.

   $[y_2 \leq x \wedge y_3 > 0 \wedge y_2 + y_3 \leq x] \Rightarrow [x - y_2 > x - (y_2 + y_3)]$

   (Naglasimo da se putanja α_1 ne razmatra zato što nije deo ni jedne petlje.)

Kako su svi uslovi u sva tri koraka tačni, na osnovu Floyd-ove teoreme o dobro zasnovanim skupovima, program se zaustavlja za svaki prirodan broj x.

3 Dijagramski programi sa nizovima

Za opisivanje velike familije promenljivih u programu koriste se nizovi. Na primer, da bi se opisala grupa od 21 celog broja, umesto korišćenja promenljivih A, B, C, ..., T, U koriste se promenljive $S[0]$, $S[1]$, ..., $S[20]$ gde je S niz od 21 elementa, a notacija odgovara matematičkoj notaciji S_0, S_1, ..., S_{20}.

3.1 Parcijalna korektnost

U ovom odeljku ćemo razmatrati parcijalnu korektnost dijagramskih programa sa nizovima i diskutovaćemo o nekim teškoćama koje se javljaju u radu sa nizovima. Ignorisaćemo jedan problem koji se tiče korektnosti dijagramskih programa sa nizovima: nećemo proveravati da li se element zaista nalazi u granicama niza. Na primer, ako pretpostavimo da program koristi niz od 21 elemenata $S[0]$, $S[1]$, ..., $S[20]$ i da u programu postoji iskaz oblika $y \leftarrow S[i + j]$ onda moramo dokazati da kad god se dođe do tog iskaza važi $0 \leq i + j \leq 20$, pri čemu su i i j unutar granica niza. Ovaj dokaz može se izvesti pridruživanjem uslova $0 \leq i + j \leq 20$ strelici koja vodi do tog stanja i njegovom proverom primenom indukcije.

Primer 4 (Bubble sort — dokaz parcijalne korektnosti) *Dokazati parcijalnu korektnost programa bubble sort za sortiranje niza $X[0]$, $X[1]$, ..., $X[n]$ od $n + 1$ realnih brojeva u rastućem poretku (slika 5).*

Prvo što treba uraditi je prepisivanje elemenata niza X u neki pomoćni niz S, zatim treba izvršiti premeštanje elemenata ovog niza S i na kraju rezultat treba upisati u niz Z koji je i rezultat rada programa. Ubuduće, operacija

$$(S[j], S[j + 1]) \leftarrow (S[j + 1], S[j])$$

će imati efekat zamene vrednosti elemenata $S[j]$ i $S[j + 1]$.
Ono što treba dokazati je da kad god se program završi, onda su ispunjeni sledeći uslovi:

1. Niz Z je permutacija elemenata originalnog niza X. To ćemo ubuduće označavati sa *perm(X,Z,0,n)*, pri čemu predikat *perm(X,Z,k,l)* označava da su elementi $Z[k]$, $Z[k+1]$, ..., $Z[l]$ permutacija elemenata $X[k]$, $X[k+1]$, ..., $X[l]$. (Po dogovoru, to je zadovoljeno ako je $k \geq l$).

2. Elementi niza Z su uređeni u rastućem poretku. Ovu osobinu ubuduće ćemo označavati sa *ordered(Z,0,n)*, pri čemu predikat *ordered(Z,k,l)* označava da su elementi $Z[k]$, $Z[k+1]$, ..., $Z[l]$ uređeni rastuće, odnosno da važi $Z[k] \leq Z[k+1] \leq, ..., \leq Z[l]$. (Po dogovoru, to je zadovoljeno ako je $k \geq l$).

Dakle, potrebno je dokazati da je program parcijalno korektan u odnosu na ulazni predikat

$$\varphi(n, X) : n \geq 0$$

i izlazni predikat

$$\psi(n, X, Z) : \textit{perm(X,Z,0,n)} \wedge \textit{ordered(Z,0,n)}.$$

Na slici 5 data je realizacija ovog algoritma.

Jasno je da je uslov *perm(X,Z,0,n)* zadovoljen, i to zbog toga što je jedina operacija koja je primenjivana operacija $(S[j], S[j+1]) \leftarrow (S[j+1], S[j])$, a ona sadržaj niza S ostavlja nepromenjenim, izuzev redosleda njegovih elemenata.

U cilju dokazivanja da je *ordered(Z,0,n)* tačno, potrebno je preseći tri petlje programa tačkama B i C i pridružiti im adekvatna induktivna tvrđenja (pogledati sliku 5). Koristićemo oznaku $\{S[k], S[k+1], ..., S[l]\}$ za skup svih elemenata $S[m]$, $k \leq m \leq l$ (ovaj skup je prazan za $k \geq l$). Za svaka dva skupa realnih brojeva T_1 i T_2, sa $T_1 \leq T_2$ označavamo da su svi elementi skupa T_1 manji ili jednaki od bilo kog elementa skupa T_2 (što je tačno i ako je neki od ovih skupova prazan).

Induktivno tvrđenje može biti iskazano na sledeći način:

- $p_1(n, X, S, i)$ u presečnoj tački B:

$$\begin{aligned} &0 \leq i \leq n \wedge \textit{ordered(S,i,n)} \\ &\wedge \{S[0], S[1], ..., S[i]\} \leq \{S[i+1], S[i+2], ..., S[n]\} \end{aligned}$$

- $p_2(n, X, S, i, j)$ u presečnoj tački C:

$$\begin{aligned} &0 \leq i \leq n \wedge 0 \leq j \leq i \wedge \textit{ordered(S,i,n)} \\ &\wedge \{S[0], S[1], ..., S[i]\} \leq \{S[i+1], S[i+2], ..., S[n]\} \\ &\wedge \{S[0], S[1], ..., S[j-1]\} \leq \{S[j]\} \end{aligned}$$

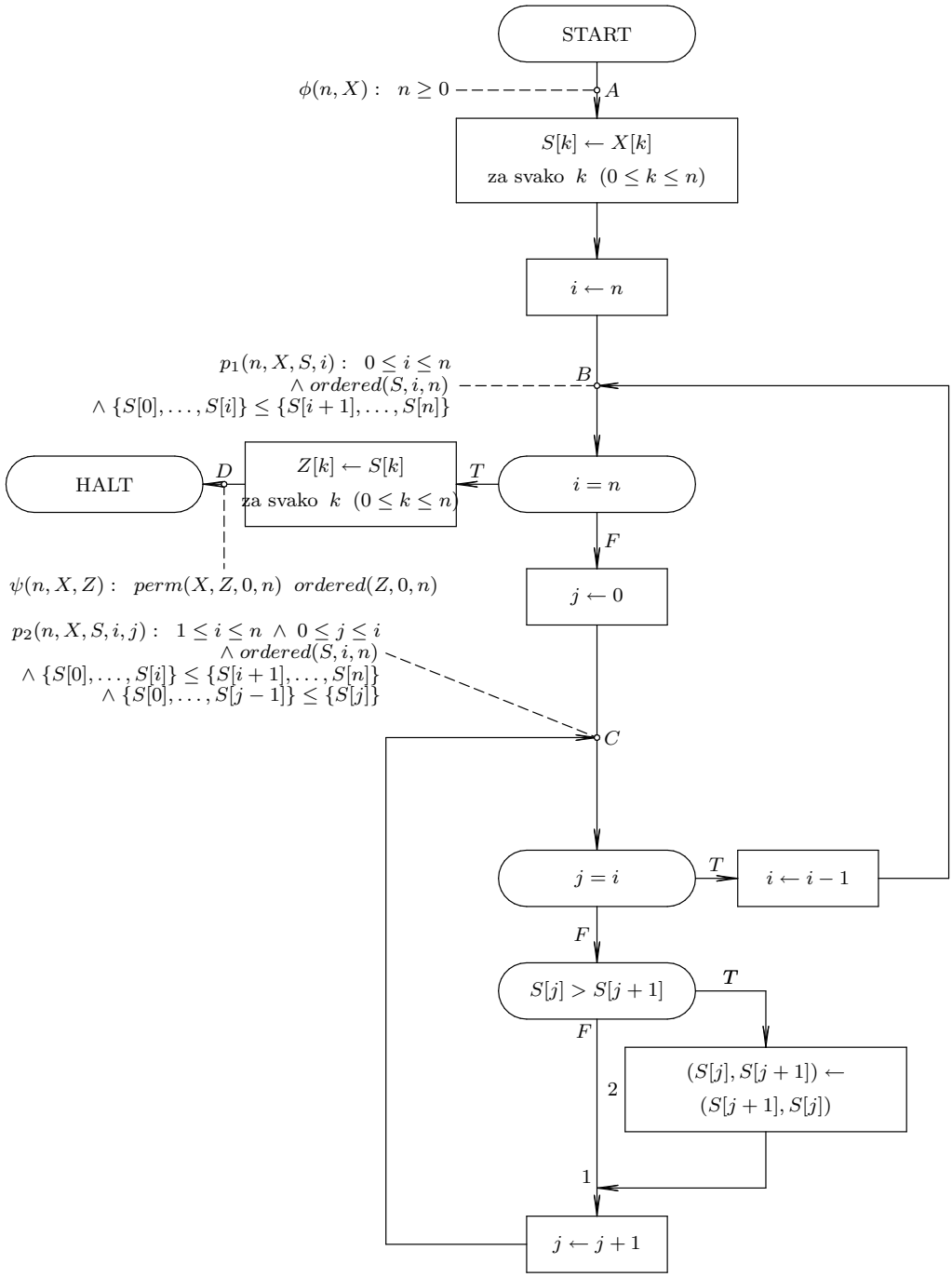
Postoji šest uslova provere koji se jednostavno dokazuju i to su:

1. Za *putanju* α (od tačke A do tačke B)

$$\varphi(n, X) \Rightarrow p_1(n, X, X, n)$$

2. Za *putanju* γ (od tačke B do tačke D)

$$[p_1(n, X, S, i) \wedge i = 0] \Rightarrow \psi(n, X, S)$$



Slika 5: Dijagramski program za sortiranje niza realnih brojeva algoritmom bubble-sort (parcijalna korektnost)

3. Za *putanju* β_1 (od tačke B do tačke C)

$$[p_1(n, X, S, i) \wedge i \neq 0] \Rightarrow p_2(n, X, S, i, 0)$$

4. Za *putanju* β_2 (od tačke C do tačke C preko strelice 1)

$$[p_2(n, X, S, i, j) \wedge (S[j] > S[j+1]) \wedge j \neq i] \Rightarrow p_2(n, X, S, i, j+1)$$

5. Za *putanju* β_3 (od tačke C do tačke C preko iskaza 2)

$$[p_2(n, X, S, i, j) \wedge (S[j] > S[j+1]) \wedge j \neq i] \Rightarrow p_2(n, X, S*, i, j+1)$$

   pri čemu S* predstavlja niz S nakon zamene vrednosti elemenata $S[j]$ i $S[j+1]$

6. Za *putanju* β_4 (od tačke C do tačke B)

$$[p_2(n, X, S, i, j) \wedge j = i] \Rightarrow p_1(n, X, S, i-1)$$

3.2 Zaustavljanje

U poglavlju 2.3 predstavljen je metod za dokazivanje zaustavljanja dijagramskih programa baziran na korišćenju dobro zasnovanih skupova. U praksi, veoma je pogodno koristiti sledeći poznati rezultat u vezi sa dobro zasnovanim skupovima pri izvođenju dokaza zaustavljanja dijagramskih programa sa nizovima.

Teorema 3 *Ako je* $(W, \prec)$ *dobro-zasnovan skup, onda je to i* $(W^n, \prec_n)$*, gde je* W^n *skup svih n-torki elemenata iz* W*, a* $\prec_n$ *je regularno leksikografsko uređenje skupa* W^n*:* $\langle a_1, a_2, ...a_n\rangle \prec_n \langle b_1, b_2, ...b_n\rangle$ *akko* $a_1 = b_1$*,* $a_2 = b_2$*, ...,* $a_{i-1} = b_{i-1}$ *i* $a_i \prec b_i$ *za neko* i $(1 \leq i \leq n)$*.*

Primer 5 (Bubble sort — dokaz zaustavljanja programa) *Dokazati da se program iz Primera 4, koji vrši sortiranje niza od* $n+1$ *realnih brojeva korišćenjem algoritma bubble-sort, zaustavlja.*

U cilju izvođenja ovog dokaza, potrebno je koristiti dobro zasnovan skup $(N^2, \prec_2)$. Potrebno je preseći tri petlje programa presečnom tačkom B i njoj pridružiti sledeće tvrđenje

$$q(n, i, j) : 0 \leq j \leq i \wedge 1 \leq i \leq n$$

i funkciju

$$u(i, j) : (i, i - j)$$

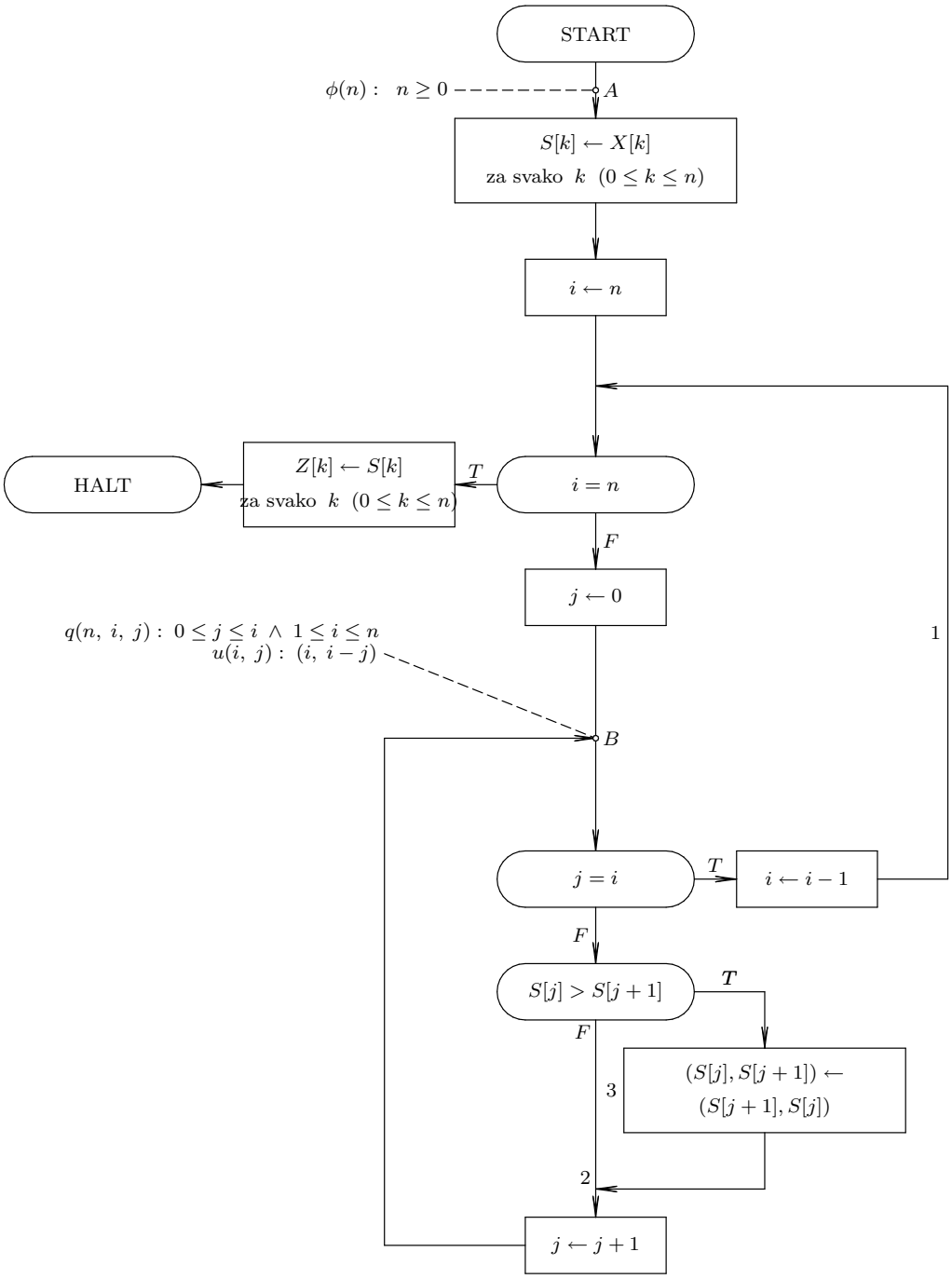
(Pogledati sliku 6).

Dokaz se izvodi u tri koraka. Treba dokazati sledeće uslove za sve vrednosti i,j i n:

1. *q je dobro tvrđenje.*

   Za putanju od tačke A do tačke B

$$[\varphi(n) \wedge n \neq 0] \Rightarrow q(n, n, 0)$$



Slika 6: Dijagramski program za sortiranje niza realnih brojeva algoritmom bubble-sort (zaustavljanje)

Za putanju od tačke B do tačke B (preko strelice 1)

$$[q(n,i,j) \wedge j = i \wedge i - 1 \neq 0] \Rightarrow q(n, i-1, 0)$$

Za putanju od tačke B do tačke B (preko strelice 2 ili iskaza 3)

$$[q(n,i,j) \wedge j \neq i] \Rightarrow q(n, i, j+1)$$

2. *u je dobra funkcija.*

$$q(n,i,j) \Rightarrow u(i,j) \in N^2$$

što se svodi na

$$[0 \leq j \leq i \wedge 1 \leq i \leq n] \Rightarrow [0 \leq i \wedge 0 \leq i - j]$$

3. *Uslovi zaustavljanja su zadovoljeni.*

Za putanju od tačke B do tačke B (preko strelice 1)

$$[q(n,i,j) \wedge j = i \wedge i - 1 \neq 0] \Rightarrow [u(i,j) >_2 u(i-1, 0)]$$

što se svodi na

$$[0 \leq j \leq i \wedge 1 \leq i \leq n \wedge j = i \wedge i - 1 \neq 0] \Rightarrow [(i, i-j) >_2 (i-1, i-1)]$$

Za putanju od tačke B do tačke B (preko strelice 2 ili iskaza 3)

$$[q(n,i,j) \wedge j \neq i] \Rightarrow [u(i,j) >_2 u(i, j+1)]$$

što se svodi na

$$[0 \leq j \leq i \wedge 1 \leq i \leq n \wedge j \neq i] \Rightarrow [(i, i-j) >_2 (i, i-(j+1))]$$

Lako se može dokazati da su svi ovi uslovi ispunjeni za sve vrednosti i,j i n, pa odatle sledi da se program zaustavlja. (Ono što se može primetiti je da dokaz zaustavljanja programa zavisi samo od vrednosti za i,j i n i ni na koji način ne zavisi od elemenata niza X.)

4 C-oliki programi

Do sada smo razmatrali klase dijagramskih programa (sa ili bez nizova) i uvideli da je suština verifikacije programa u načinu razmatranja petlji. U praksi, programi se pišu u linearnoj formi a petlje se mogu izraziti na veliki broj načina. U ovom poglavlju diskutovaćemo verifikaciju klase *while programa*, programa u kojima su petlje izražene u terminima WHILE petlje.

4.1 While programi

Definicija 8 While program *se sastoji od konačnog niza naredbi B_i razdvojenih tačka-zarezima:*

$$B_0; B_1; B_2; \ldots; B_n$$

gde je B_0 jedinstvena START naredba oblika

$$\texttt{START}$$
$$\bar{y} \leftarrow f(\bar{x})$$

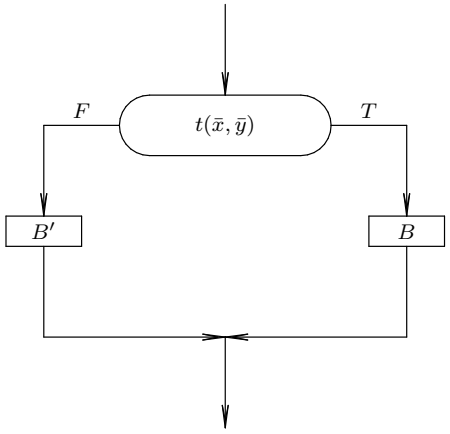
a svako B_i*, za* $1 \leq i \leq n$*, je jedna od sledećih naredbi:*

1. *Naredba DODELE:*
$$\bar{y} \leftarrow g(\bar{x}, \bar{y})$$
2. *naredba USLOVA:*
$$\texttt{if } t(\bar{x}, \bar{y}) \texttt{ then } B \texttt{ else } B^{'}$$
ili
$$\texttt{if } t(\bar{x}, \bar{y}) \texttt{ do } B$$
gde su B *i* $B^{'}$ *proizvoljne naredbe.*
3. *WHILE naredba:*
$$\texttt{while } t(\bar{x}, \bar{y}) \texttt{ do } B$$
gde je B *proizvoljna naredba.*
4. *HALT naredba:*
$$\bar{z} \leftarrow h(\bar{x}, \bar{y})$$
$$\texttt{HALT}$$
5. *BLOK naredba:*
$$\texttt{begin } B_1^{'}; B_2^{'}; \ldots ; B_k^{'} \texttt{ end}$$
gde je $B_j^{'}$ *za* $1 \leq j \leq k$ *proizvoljna naredba.*

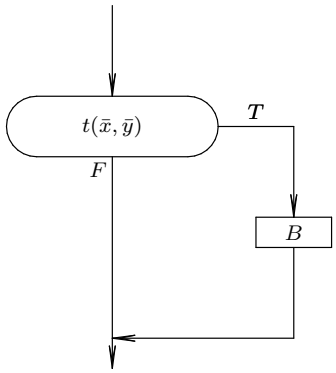
Ako je dat *while program* P i ulazna vrednost $\bar{\xi} \in D_{\bar{x}}$ za ulazni vektor $\bar{x}$, tada program može da se izvršava. Izvršavanje programa počinje naredbom START, $\bar{y}$ se inicijalizuje vrednošću $f(\bar{\xi})$ i nastavlja se na normalan način, prateći naredbe programa. Kad god se dođe do naredbe DODELE, $\bar{y}$ se zamenjuje tekućom vrednošću funkcije $g(\bar{x}, \bar{y})$. Izvršavanje USLOVNE i WHILE naredbe može da se opiše dijagramima toka programa prikazanim na slikama 7, 8, 9).

Primer 6 (Celobrojni koren) *Program koji izračunava celobrojni koren prirodnog broja iz primera 2 može se napisati na sledeći način:*
START
$(y_1, y_2, y_3) \leftarrow (0, 1, 1)$;
while $y_2 \leq x$ do $(y_1, y_2, y_3) \leftarrow (y_1 + 1, y_2 + y_3 + 2, y_3 + 2)$;
$z \leftarrow y_1$
HALT

Primer 7 (NZD) *Program koji izračunava najveći zajednički delilac dva prirodna broja može se napisati na sledeći način:*
START $(y_1, y_2) \leftarrow (x_1, x_2)$;
while $y_1 \neq y_2$ do if $y_1 > y_2$ then $y_1 \leftarrow y_1 - y_2$ else $y_2 \leftarrow y_2 - y_1$;
$z \leftarrow y_1$
HALT



Slika 7: if $t(\bar{x}, \bar{y})$ then B else $B^{'}$



Slika 8: if $t(\bar{x}, \bar{y})$ do B

4.2 Parcijalna korektnost

U svrhu dokazivanja parcijalne korektnosti *while programa* uvodi se sledeća oznaka:

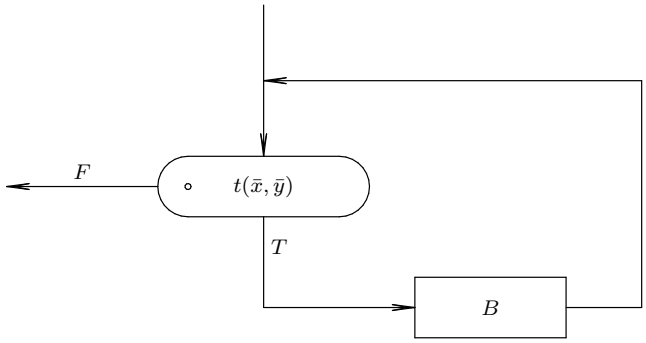
$$\{p(\bar{x}, \bar{y})\}B\{q(\bar{x}, \bar{y})\}$$

gde su p i q predikati a B deo programa. Ovaj zapis označava da ako važi $p(\bar{x}, \bar{y})$ za vrednosti $\bar{x}$ i $\bar{y}$ neposredno pre izvršavanja B, tada ako se B završi onda će $q(\bar{x}, \bar{y})$ da važi za $\bar{x}$ i $\bar{y}$ nakon izvršavanja B.

Ako želimo da dokažemo parcijalnu korektnost za dati *while program* P u odnosu na ulazni predikat φ i izlazni predikat ψ, dovoljno je dokazati da važi:

$$\{\varphi(\bar{x})\}P\{\psi(\bar{x}, \bar{z})\}.$$

U tu svrhu koristimo *verifikaciona pravila* koja se sastoje od *aksiome dodele* i od *pravila izvođenja*. Aksioma dodele opisuje transformacije programskih promenljivih prilikom izvršavanja naredbe DODELE. Pravila izvođenja omo-



Slika 9: while $t(\bar{x}, \bar{y})$ do B

gućavaju izraze manjih delova programa da se kombinuju u izraze koje obuhvataju veće delove programa. Da bi mogla da se primenjuju pravila, prvo se izvode izrazi oblika $\{p\}B\{q\}$ za svaku naredbu DODELE B programa, koristeći aksiomu dodele. Dalje se primenjuju pravila izvođenja dok se ne dođe do željenog izraza $\{\varphi\}P\{\psi\}$.

Pravila izvođenja se opisuju na sledeći način:

$$\frac{\alpha_1}{\beta} \text{ ili } \frac{\alpha_1 \text{ i } \alpha_2}{\beta}$$

gde su α_1 i α_2 *preduslovi* tj. uslovi pod kojima pravilo može da se primeni, a β *posledica* tj. izraz koji se izvodi dedukcijom. Svaki preduslov je ili izveden izraz koji je ranije utvrđen ili logički izraz koji treba da bude dokazan zasebno u obliku leme.

Verifikaciona pravila su:

1. Aksioma *dodele*:

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

2. *Uslovna* pravila:

$$\frac{\{p \wedge t\}B_1\{q\} \text{ i } \{p \wedge \neg t\}B_2\{q\}}{\{p\} \text{ if } t \text{ then } B_1 \text{ else } B_2\{q\}}$$

   i

$$\frac{\{p \wedge t\}B\{q\} \text{ i } \{p \wedge \neg t\} \Rightarrow q}{\{p\} \text{ if } t \text{ do } B\{q\}}$$

   Napomena: formula $\{p \wedge \neg t\} \Rightarrow q$ je logički izraz koji treba shvatiti kao dobro zasnovanu logičku formulu gde su sve njene promenljive implicitno vezane univerzalnim kvantifikatirima, tj:

$$\forall\bar{x}\forall\bar{y}[\{p(\bar{x}, \bar{y}) \wedge \neg t(\bar{x}, \bar{y})\} \Rightarrow q(\bar{x}, \bar{y})]$$

3. *While* pravilo:

$$\frac{\{p \wedge t\}B\{p\}}{\{p\} \text{ while } t \text{ do } B\{p \wedge \neg t\}}$$

Pravilo za WHILE naredbu se zasniva na činjenici da ako se izvrši telo petlje B, onda WHILE naredba ostavlja tvrdnju p invarijantnom (tj. p je uvek tačno kada se izvrše naredbe B, pri čemu takođe važi da je p tačno i prilikom inicijalizacije), pa je p tačno posle proizvoljnog broja iteracija izvršavanja B.

4. Pravilo *nadovezivanja*:

$$\frac{\{p\}B_1\{q\} \text{ i } \{p\}B_2\{r\}}{\{p\}B_1;\ B_2\{r\}}$$

5. *Posledično* pravilo:

$$\frac{\{p\}B\{q\} \text{ i } q \Rightarrow r}{\{p\}B\{r\}}$$

i

$$\frac{p \Rightarrow q \text{ i } \{q\}B\{r\}}{\{p\}B\{r\}}$$

Primetimo ponovo da su $p \Rightarrow q$ i $q \Rightarrow r$ logički izrazi koji treba da su dokazani zasebno kao leme.

Teorema 4 (Metod verifikacionih pravila (Hoare)) *Neka su dati* while program P, *ulazni predikat* $\varphi(\bar{x})$ *i izlazni predikat* $\psi(\bar{y})$. *Ako se primenom verifikacionih pravila može izvesti*

$$\{\varphi(\bar{x})\}P\{\psi(\bar{x},\bar{z})\}$$

tada je P *parcijalno korektan program u odnosu na* φ *i* ψ.

Verifikaciona pravila mogu da se kombinuju tako da formiraju nova pravila koja su pogodnija za upotrebu. Ta pravila se obično zovu *izvedena verifikaciona pravila*. Neka od njih su:

1. *Pravilo dodele* (dobija se iz aksiome dodele i posledičnog pravila):

$$\frac{p(\bar{x},\bar{y}) \Rightarrow q(\bar{x},g(\bar{x},\bar{y}))}{\{p(\bar{x},\bar{y})\}\bar{y} \leftarrow g(\bar{x},\bar{y})\{q(\bar{x},\bar{y})\}}$$

2. *Pravilo dodele sa ponavljanjima* (dobija se iz pravila dodela i pravila nadovezivanja):

$$\frac{p(\bar{x},\bar{y}) \Rightarrow q(\bar{x},g_n(\bar{x},g_{n-1}(\bar{x},...,g_2(\bar{x},g_1(\bar{x},\bar{y}))...)))}{\{p(\bar{x},\bar{y})\}\bar{y} \leftarrow g(\bar{x},\bar{y})\{q(\bar{x},\bar{y})\}}$$

3. *Modifikovano while pravilo* (dobija se iz while pravila i posledičnog pravila):

$$\frac{\{p \wedge t\}B\{q\} \text{ i } \{p \wedge \neg t\} \Rightarrow q}{\{p\} \text{ while } t \text{ do } B\{q\}}$$

4. *Modifikovano posledično pravilo* (dobija se iz pravila nadovezivanja i posledičnog pravila):

$$\frac{\{p\}B_1\{q\} \text{ i } \{r\}B_2\{s\} \text{ i } q \Rightarrow r}{\{p\}B_1;\ B_2\{s\}}$$

Primer 8 (Celobrojni koren — dokaz parcijalne korektnosti) *Dokažimo parcijalnu korektnost programa 6 za izračunavanje celobrojnog korena prirodnog broja u odnosu na ulazni predikat* $\varphi(x) : x \geq 0$ *i izlazni predikat* $\psi(x, z) : z^2 \leq x \leq (z+1)^2$.

Program je:

```
START
(y1, y2, y3) ← (0, 1, 1);
while y2 ≤ x do (y1, y2, y3) ← (y1 + 1, y2 + y3 + 2, y3 + 2);
z ← y1
HALT
```

U dokazu ćemo koristiti sledeći predikat:

$$R(x, y_1, y_2, y_3)\text{: } (y_1^2 \leq x) \wedge (y_2 = (y_1+1)^2) \wedge (y_3 = 2y_1 + 1)$$

Dokaz se može izvesti formalno na sledeći način:

1.	$x \geq 0 \Rightarrow R(x, 0, 1, 1)$	Lema 1
2.	$\{x \geq 0\}$ $(y_1, y_2, y_3) \leftarrow (0, 1, 1)$ $\{R(x, y_1, y_2, y_3)\}$	Pravilo dodele (linija 1)
3.	$R(x, y_1, y_2, y_3) \wedge y_2 \leq x \Rightarrow R(y_1 + 1, y_2 + y_3 + 2, y_3 + 2)$	Lema 2
4.	$\{R(x, y_1, y_2, y_3) \wedge y_2 \leq x\}$ $(y_1, y_2, y_3) \leftarrow (y_1 + 1, y_2 + y_3 + 2, y_3 + 2)$ $\{R(x, y_1, y_2, y_3)\}$	Pravilo dodele(linije 4)
5.	$\{R(x, y_1, y_2, y_3)\}$ `while` $y_2 \leq x$ `do` $(y_1, y_2, y_3) \leftarrow (y_1 + 1, y_2 + y_3 + 2, y_3 + 2)$ $\{R(x, y_1, y_2, y_3) \wedge y_2 > x\}$	while pravilo(linija 4)
6.	$\{x \geq 0\}$ $(y_1, y_2, y_3) \leftarrow (0, 1, 1)$ `while` $y_2 \leq x$ `do` $(y_1, y_2, y_3) \leftarrow (y_1 + 1, y_2 + y_3 + 2, y_3 + 2)$ $\{R(x, y_1, y_2, y_3) \wedge y_2 > x\}$	Pravilo nadovezivanja (linije 2 i 5)
7.	$R(x, y_1, y_2, y_3) \wedge y_2 > x \Rightarrow y_1^2 \leq x < (y_1+1)^2$	Lema 3
8.	$\{R(x, y_1, y_2, y_3) \wedge y_2 > x\}$ $z \leftarrow y_1$ $\{z^2 \leq x < (z+1)^2\}$	Pravilo dodele (linija 7)
9.	$\{x \geq 0\}$ $(y_1, y_2, y_3) \leftarrow (0, 1, 1)$ `while` $y_2 \leq x$ `do` $(y_1, y_2, y_3) \leftarrow (y_1 + 1, y_2 + y_3 + 2, y_3 + 2)$ $z \leftarrow y_1$ $\{z^2 \leq x < (z+1)^2\}$	Pravilo nadovezivanja (linije 6 i 8)

Dakle, s obzirom na to da su leme 1, 2 i 3 tačne, navedeni dokaz povlači da je program parcijalno korektan u odnosu na ulazni predikat $x \geq 0$ i izlazni predikat $z^2 \leq x \leq (z+1)^2$.

Kao i kod metoda iduktivnog tvrđenja, izbor predikata je suštinski deo dokaza. Proces određivanja pogodnog predikata zahteva duboko razumevanje svojstava programa. Bilo bi dobro ako bi to bilo obezbeđeno od strane programera za svaki program koji piše, i to u vidu komentara na odgovarajućim

mestima u kodu. Na primer, prikladan način za prikazivanje pograma za celobrojno deljenje bi bio (komentari su navedeni u zagradama):

```
START
{x ≥ 0}
(y1, y2, y3) ← (0, 1, 1);
while y2 ≤ x do {(y1^2 ≤ x) ∧ (y2 = (y1 + 1)^2) ∧ (y3 = 2y1 + 1)}
(y1, y2, y3) ← (y1 + 1, y2 + y3 + 2, y3 + 2);
z ← y1
{z^2 ≤ x < (z + 1)^2}
HALT
```

4.3 Totalna korektnost

Prethodna pravila omogućavaju dokazivanje samo parcijalne korektnosti *while programa*. Sa ciljem da proširimo metod radi dokazivanje totalne korektnosti, uvodimo sledeću notaciju:

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

sa značenjem da za svako $\bar{x}$ i $\bar{y}$ ako važi $p(\bar{x}, \bar{y})$ neposredno pre izvršavanja B, izvršavanje programa B će se završiti i važiće $q(\bar{x}, \bar{y}, \bar{y}')$ gde je $\bar{y}'$ skup rezultujućih vrednosti za $\bar{y}$. Kako se $\bar{x}$ ne menja u toku izvršavanja progama, umesto $p(\bar{x}, \bar{y})$ i $q(\bar{x}, \bar{y}, \bar{y}')$ pisaćemo kraće $p(\bar{y})$ i $q(\bar{y}, \bar{y}')$.

Nova verifikaciona pravila zaustavljanja su:

1. *Pravilo dodele:*
$$\frac{\forall\bar{y}\forall\bar{y}'[p(\bar{y}) \wedge \bar{y}' = f(\bar{y}) \Rightarrow q(\bar{y}, \bar{y}')]}{\{p(\bar{y})\}\bar{y} \leftarrow f(\bar{y})\{q(\bar{y}, \bar{y}')\}}$$

2. *Uslovna pravila:*
$$\frac{\{p(\bar{y}) \wedge t(\bar{y})\}B_1\{q(\bar{y}, \bar{y}')\} \quad \{p(\bar{y}) \wedge t(\bar{y})\}B_2\{q(\bar{y}, \bar{y}')\}}{\{p(\bar{y})\} \text{ if } t(\bar{y}) \text{ then } B_1 \text{ else } B_2\{q(\bar{y}, \bar{y}')\}}$$
i
$$\frac{\forall\bar{y}\forall\bar{y}'[p(\bar{y}) \wedge \neg t(\bar{y}) \Rightarrow q(\bar{y}, \bar{y}')]}{\{p(\bar{y})\} \text{ if } t(\bar{y}) \text{ do } B\{q(\bar{y}, \bar{y}')\}}$$

3. *Pravilo nadovezivanja:*
$$\frac{\{p_1(\bar{y})\}B_1\{q_1(\bar{y}, \bar{y}')\} \quad \{p_2(\bar{y})\}B_2\{q_2(\bar{y}, \bar{y}')\} \quad \forall\bar{y}\forall\bar{y}'[q_1(\bar{y}, \bar{y}') \Rightarrow p_2(\bar{y}')] \quad \forall\bar{y}\forall\bar{y}'\forall\bar{y}''[q_1(\bar{y}, \bar{y}') \wedge q_2(\bar{y}', \bar{y}'') \Rightarrow q(\bar{y}, \bar{y}'')]}{\{p_1(\bar{y})\}B_1;\ B_2\{q(\bar{y}, \bar{y}')\}}$$

4. *Posledično pravilo:*

$$\frac{\{r(\bar{y})\}B\{q(\bar{y},\bar{y}')\} \quad \forall\bar{y}[p(\bar{y}) \Rightarrow r(\bar{y})]}{\{p(\bar{y})\}B\{q(\bar{y},\bar{y}')\}}$$

i

$$\frac{\{p(\bar{y})\}B\{s(\bar{y},\bar{y}')\} \quad \forall\bar{y}\forall\bar{y}'[s(\bar{y}) \Rightarrow q(\bar{y})]}{\{p(\bar{y})\}B\{q(\bar{y},\bar{y}')\}}$$

5. *Ili pravilo:*

$$\frac{\{p_1(\bar{y})\}B\{q(\bar{y},\bar{y}')\} \quad \{p_2(\bar{y})\}B\{q(\bar{y},\bar{y}')\}}{\{p_1(\bar{y}) \vee p_2(\bar{y})\}B\{q(\bar{y},\bar{y}')\}}$$

6. I pravilo:

$$\frac{\{p(\bar{y})\}B\{q_1(\bar{y},\bar{y}')\} \quad \{p(\bar{y})\}B\{q_2(\bar{y},\bar{y}')\}}{\{p(\bar{y})\}B\{q_1(\bar{y},\bar{y}') \wedge q_2(\bar{y},\bar{y}')\}}$$

7. *While pravilo:*

$$\frac{\begin{array}{c}\{p(\bar{y}) \wedge t(\bar{y})\}B\{q(\bar{y},\bar{y}') \wedge [u(\bar{y}) \prec u(\bar{y}')]\} \\ \forall\bar{y}\forall\bar{y}'[q_1(\bar{y},\bar{y}') \wedge t(\bar{y}') \Rightarrow p(\bar{y}')] \\ \forall\bar{y}\forall\bar{y}'\forall\bar{y}''[q_1(\bar{y},\bar{y}') \wedge q_2(\bar{y}',\bar{y}'') \Rightarrow q(\bar{y},\bar{y}'')] \\ \forall\bar{y}[p(\bar{y}) \wedge \neg t(\bar{y}) \Rightarrow q(\bar{y},\bar{y}')]\end{array}}{\{p(\bar{y})\} \text{ while } t(\bar{y}) \text{do } B \ \{q(\bar{y},\bar{y}'') \wedge \neg t(\bar{y}')\}}$$

gde je $(W, \prec)$ dobro zasnovan skup i $u : D_{\bar{x}} \times D_{\bar{y}} \to W$.

Teorema 5 (Metod verifikacionih pravila zaustavljanja (Manna i Pnueli)) *Neka su dati* while program P*, ulazni predikat* $\varphi(\bar{x})$ *i izlazni predikat* $\psi(\bar{y})$*. Ako se primenom verifikacionih pravila zaustavljanja može izvesti*

$$\{\varphi(\bar{x})\}P\{\psi(\bar{x},\bar{z})\}$$

tada je P *totalno korektan program u odnosu na* φ *i* ψ*.*