SEMINAR 05 – VVSS – 27.04.2020, 2019-2020

Studiați parțial corectitudinea, terminarea și total corectitudinea următorilor algoritmi:

1) cmmdc a două numere naturale

$\varphi(X): n_1, n_2 \in \mathbb{N}$

$\psi(X,Z): d = cmmdc(n_1, n_2)$

$X=(n_1, n_2)$; $Z=(d)$

$Y=(d,i) \rightarrow$ vector cu variabile ce conțin rezultate intermediare

```
int cmmdc(int n1, int n2) {
   • A: [φ(X): n1,n2∈N]
   int d=n1, i=n2;
   while ((d!=i) && (i>0)) {
       • B: [P_B(X,Y): cmmdc(d,i)=cmmdc(n1,n2)]
       if (d>i)
           d=d-i;
       else i=i-d;
   }
   return d;
   • C: [ψ(X,Z): d=cmmdc(n1,n2)]
}
```

I $\boxed{\text{Parțial corectitudine}}$ (vezi curs 07)

1) identificarea punctelor de tăietură: A, B, C

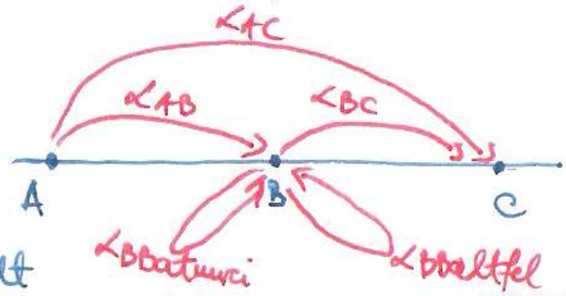
2) identificarea predicatelor invariante asociate punctelor de tăietură:

$\varphi(X)$, $\psi(X,Z)$, $P_B(X,Y)$

3) identificarea drumurilor α_{ij}:

5 drumuri: α_{AB}, α_{BC}, α_{AC}, α_{BB}

α_{BBat} α_{BBalt}



4) identificarea condițiilor de parcurgere a drumurilor α_{ij} ($R_{\alpha ij}(X,Y)$): ②

$R_{\alpha ij}(X,Y)$ = predicat

- 5 drumuri => 5 predicate $R_{\alpha ij}(X,Y)$

• $R_{\alpha AB}(X,Y) = (d \neq i) \wedge (i > 0)$ ⟿ condiția de intrare în "while"

• $R_{\alpha DBat}(X,Y) = (d \neq i) \wedge (i > 0) \wedge \underbrace{(d > i)}$

condiția de execuție încă o dată a buclei, pornind din B și ajungând din nou în B, pe ramura "then" a instrucțiunii "if"

• $R_{\alpha DBatt}(X,Y) = (d \neq i) \wedge (i > 0) \wedge \underbrace{(d \leq i)}$

condiția de parcurgere a ramurii "else" din instrucțiunea "if"

• $R_{\alpha BC}(X,Y) = ![(d \neq i) \wedge (i > 0)] = (d = i) \vee (i \leq 0) = (d = i) \vee (i = 0)$

condiția de ieșire din "while"

$i < 0$? $\left.\begin{array}{l} i = n2 \\ n2 \in \mathbb{N} \end{array}\right\} \Rightarrow n2 \geq 0 \Rightarrow i \geq 0$ $\left.\begin{array}{l} \\ \text{dar, } i \leq 0 \end{array}\right\} \Rightarrow i = 0$

• $R_{\alpha AC}(X,Y) = R_{\alpha BC}(X,Y) = (d = i) \vee (i = 0)$

condiția de pornire din A și de sosire în C, fără a intra în "while"

5) identificarea transformărilor pe drumurile α_{ij} ($r_{\alpha ij}(X,Y)$):

$r_{\alpha ij}(X,Y)$ = funcție ≠ predicat

- 5 drumuri => 5 funcții de transformare $\alpha_{ij}(X,Y)$:

• $r_{\alpha AB}(X,Y) = (d, i) = (n1, n2)$;

pe drumul α_{AB} variabilele d și i își modifică valorile, adică sunt inițializate cu n1 și n2

- $r_{\alpha_{BB}at}(X,Y) = (d,i) = (d-i,i)$

$\Downarrow$

pe drumul α_{BB} atunci doar d își modifică valoarea

- $r_{\alpha_{BB}altfel}(X,Y) = (d,i) = (d,i-d)$

$\Downarrow$

pe drumul α_{BB}altfel doar i își modifică valoarea

- $r_{\alpha_{BC}}(X,Y) = (d,i) = \begin{cases} (d-i,i), & d>i \\ (d,i-d), & d\le i \end{cases}$

$\Downarrow$

pe drumul α_{BC} se poate modifica fie d, fie i, în funcție de îndeplinirea condiției "d>i"

- $r_{\alpha_{AC}}(X,Y) = (d,i) = (n1,n2) = r_{\alpha_{AB}}(X,Y)$

$\Downarrow$

pe drumul α_{AC} au loc aceleași transformări ce au loc pe α_{AB}, deoarece nu se intră în "while"

6) construirea și demonstrarea condițiilor de verificare asociate α_{ij}:

$$V_{\alpha_{ij}}: \forall X, \forall Y [P_i(X,Y) \wedge R_{\alpha_{ij}}(X,Y) \rightarrow P_j(X, r_{\alpha_{ij}}(X,Y))]$$

- cum "citim" condiția de verificare? oricare ar fi vectorul X și Y, dacă pornim execuția din punctul de tăietură i, în care este adevărat predicatul invariant $P_i(X,Y)$ și parcurgem drumul α_{ij} pentru care este satisfăcută condiția de parcurgere $R_{\alpha_{ij}}(X,Y)$, ajungem în punctul de tăietură j, în care este adevărat predicatul invariant $P_j(X,Y)$, iar asupra variabilelor au avut loc transformările indicate de funcția $r_{\alpha_{ij}}(X,Y)$

- 5 drumuri => se construiesc și se demonstrează 5 condiții de verificare, asociate drumurilor α_{ij}

• $V_{\angle AB}: \forall X \forall Y \left[P_A(X,Y) \wedge R_{\angle AB}(X,Y) \rightarrow P_B(X, r_{\angle AB}(X,Y)) \right]$ $P_A \equiv \varphi$

$(n_1, n_2 \in \mathbb{N}) \wedge ((d \neq i) \wedge (i > 0)) \xrightarrow{r_{\angle AB}} cmmdc(d,i) = cmmdc(n_1, n_2)$

$r_{\angle AB}(X,Y) = (d,i) = (n_1, n_2) \Rightarrow cmmdc(n_1, n_2) = cmmdc(n_1, n_2)$ „a"

• $V_{\angle BBat}: \forall X, \forall Y \left[P_B(X,Y) \wedge R_{\angle BBat}(X,Y) \rightarrow P_B(X, r_{\angle BBat}(X,Y)) \right]$

$x = cmmdc(d,i) = cmmdc(n_1, n_2) \wedge (d \neq i) \wedge (i > 0) \wedge (d > i) \xrightarrow{r_{\angle BBat}}$

$cmmdc(\underset{d-i}{d}, i) = cmmdc(n_1, n_2)$

$r_{\angle BBat}(X,Y) = (d,i) = (d-i, i)$

$$\left.\begin{array}{l} x = cmmdc(d,i) = cmmdc(n_1, n_2) \\ x = cmmdc(d-i, i) \end{array}\right\} \Rightarrow cmmdc(d-i, i) = cmmdc(n_1, n_2) = x$$

„a"

• $V_{\angle BBalt}: \forall X, \forall Y \left[P_B(X,Y) \wedge R_{\angle BBalt}(X,Y) \rightarrow P_B(X, r_{\angle BBalt}(X,Y)) \right]$

$x = cmmdc(d,i) = cmmdc(n_1, n_2) \wedge (d \neq i) \wedge (i > 0) \wedge (d \leq i) \xrightarrow{r_{\angle BBalt}}$

$cmmdc(d, \underset{i-d}{i}) = cmmdc(n_1, n_2)$

$r_{\angle BBalt}(X,Y) = (d,i) = (i, i-d)$

$$\left.\begin{array}{l} x = cmmdc(d,i) = cmmdc(n_1, n_2) \\ x = cmmdc(d, i-d) \end{array}\right\} \Rightarrow cmmdc(d, i-d) = cmmdc(n_1, n_2) = x$$

„a"

• $V_{\angle BC}: \forall X, \forall Y \left[P_B(X,Y) \wedge R_{\angle BC}(X,Y) \rightarrow P_C(X, r_{\angle BC}(X,Y)) \right]$ $P_C \equiv \psi$

$cmmdc(d,i) = cmmdc(n_1, n_2) \wedge ((d = i) \vee (i = 0)) \xrightarrow{r_{\angle BC}} d = cmmdc(n_1, n_2)$

1) $d = i \Rightarrow cmmdc(d,i) = cmmdc(d,d) = d \Rightarrow d = cmmdc(n_1, n_2)$ „a"

2) $i = 0 \Rightarrow cmmdc(d,i) = cmmdc(d,0) = d \Rightarrow d = cmmdc(n_1, n_2)$ „a"

• $V_{\angle AC}: \forall X, \forall Y [P_A(X,Y) \wedge R_{\angle AC}(X,Y) \longrightarrow P_C(X, r_{\angle AC}(X,Y))]$ $\quad P_A \equiv \varphi$, $P_C \equiv \psi$

$n_1, n_2 \in \mathbb{N} \wedge ((d=i) \vee (i=0)) \xrightarrow{r_{\angle AC}} d = \text{cmmdc}(n_1, n_2)$

1) $d=i \Rightarrow d = n_1, i = n_2 \Rightarrow n_1 = n_2 = d \Rightarrow d = \text{cmmdc}(n_1, n_2)$ „a"

$r_{\angle AC}(X,Y) = (d,i) = (n_1, n_2)$

2) $i=0 \Rightarrow d = n_1, i = 0 \Rightarrow d = \text{cmmdc}(d, 0)$ „a"

- pe baza teoremei (vezi curs 07) ⇒ algoritmul este parțial corect în raport cu specificația $(\varphi(X), \psi(X,Z))$

V. **Terminare** (vezi curs 07)

- pașii 1 → 5 sunt similari

6) se alege o mulțime convenabilă M, pentru care:
   - există ordine parțială
   - nu există șiruri descrescătoare infinite

   M poate fi $\mathbb{N}$

7) se definește pentru fiecare punct de tăietură o funcție

$u_i : D_X \times D_Y \to M$

$u_A(X,Y) = 2(n_1 + n_2)$ ... sau $n_1 + n_2 + 1$

$u_B(X,Y) = d + i$

$u_C(X,Y) = 0$

descrește, avem o secvență finită de pași

8) construirea și demonstrarea condițiilor de terminare asociate drumurilor α_{ij}:

$T_{\alpha ij}: \forall X, \forall Y [P_i(X,Y) \wedge R_{\alpha ij}(X,Y) \to u_i(X,Y) > u_j(X, r_{\alpha ij}(X,Y))]$

„citim" condiția de terminare: oricare ar fi X și Y, pornind din punctul de tăietură i, în care este adevărat predicatul invariant $P_i(X,Y)$ și parcurgem drumul α_{ij} pentru care este satisfăcută condiția de parcurgere $R_{\alpha ij}(X,Y)$, valoarea funcției u, descrește de la punctul i la punctul j atunci când au avut loc transformările indicate de funcția $r_{\alpha ij}(X,Y)$ ⑥

-5 drumuri => se construiește și se demonstrează 5 condiții de terminare asociate drumurilor α_{ij}

• $T_{\alpha AB}: \forall X, \forall Y\ [P_A(X,Y) \wedge R_{\alpha AB}(X,Y) \rightarrow u_A(X,Y) > u_B(X, r_{\alpha AB}(X,Y))]$

$n_1, n_2 \in N \wedge (d \neq i) \wedge (i > 0) \xrightarrow{r_{\alpha AB}} 2(n_1+n_2) > d+i$

$r_{\alpha AB}(X,Y) = (d,i) = (n_1, n_2)$ $\qquad 2(n_1+n_2) > n_1+n_2$ „a"

• $T_{\alpha BBat}: \forall X, \forall Y\ [P_B(X,Y) \wedge R_{\alpha BBat}(X,Y) \rightarrow u_B(X,Y) > u_B(X, r_{\alpha BBat}(X,Y))]$

$cmmdc(d,i) = cmmdc(n_1,n_2) \wedge (d \neq i) \wedge (i>0) \wedge (d>i) \xrightarrow{r_{\alpha BBat}} d+i > d+i$

$r_{\alpha BBat}(X,Y) = (d,i) = (d-i, i)$ $\qquad \cancel{d}+i > (\cancel{d}-\cancel{i})+\cancel{i}$

$i > 0$ „a"

• $T_{\alpha BBalt}: \forall X, \forall Y\ [P_B(X,Y) \wedge R_{\alpha BBalt}(X,Y) \rightarrow u_B(X,Y) > u_B(X, r_{\alpha BBalt}(X,Y))]$

$cmmdc(d,i) = cmmdc(n_1,n_2) \wedge \underbrace{(d \neq i) \wedge (i>0) \wedge (d \leq i)}_{0 \leq d < i} \xrightarrow{r_{\alpha BBalt}} d+i > d+i$

$r_{\alpha BBalt} = (d,i) = (i, i-d)$ $\qquad d+\cancel{i} > \cancel{d} + (\cancel{i}-\cancel{d})$

$d > 0$

contraexemplu: pt $n_1 = 0, n_2 \neq 0 \Rightarrow d = 0 \Rightarrow T_{\alpha BBalt}$ – false, adică se intră într-o buclă infinită; se execută de o infinitate de ori $r_{\alpha BBalt}$, unde d nu își modifică valoarea => trebuie modificat algoritmul a.î. drumul α_{BBalt} să nu se parcurgă de o infinitate de ori

– o posibilă modificare, inițializarea variabilelor d și i să se realizeze condiționat:

```
if (n1 = 0) { d = n2; i = 0; }
else { d = n1; i = n2; }
```

- pentru $u_1 \neq 0, u_2 \neq 0 \Rightarrow T_{\angle BBalt}$ true / adevărat

• $T_{\angle BC}: \forall X, \forall Y\, [P_B(X,Y) \wedge R_{\angle BC}(X,Y) \rightarrow u_B(X,Y) > u_C(X, r_{\angle BC}(X,Y))]$

$$cmmdc(d,i) = cmmdc(u_1,u_2) \wedge ((d=i) \vee (i=0)) \xrightarrow{r_{\angle BC}} d+i>0$$

$$r_{\angle BC}(X,Y) = \begin{cases} (d,i)=(d-i,i), & d>i \\ (d,i)=(d,i-d), & d\leq i \end{cases}$$

1) $d>i$, $d+i>0$

• $d=i$, contradicție cu $d>i$

• $i=0$, $d+0>0 \Rightarrow d>0$ „a"

2) $d\leq i$, $d+i>0$

• $d=i$, $d+d>0 \Rightarrow d>0$ „a"

• $i=0$, $d+0>0$, dar, $d\leq i$, $i=0$ $\Big\} \Rightarrow d\leq 0$ – contradicție

• $T_{\angle AC}: \forall X, \forall Y\, [P_A(X,Y) \wedge R_{\angle AC}(X,Y) \rightarrow u_A(X,Y) > u_C(X, r_{\angle AC}(X,Y))]$

$$u_1,u_2 \in \mathbb{N} \wedge ((d=i) \vee (i=0)) \xrightarrow{r_{\angle AC}} 2(u_1+u_2)>0$$

$r_{\angle AC}(X,Y) = (d,i) = (u_1,u_2)$

1) $d=i$, $2(u_1+u_2)>0$

$\Rightarrow u_1=u_2 \Rightarrow 2\cdot 2u_1>0 \Rightarrow u_1>0$? întotdeauna?

2) $i=0$, $2(u_1+u_2)>0$

$\Rightarrow d=u_1, i=u_2=0 \Rightarrow 2(u_1+0)>0 \Rightarrow u_1>0$? întotdeauna?

- demonstrarea $T_{\angle BC}$ și $T_{\angle AC}$ nu sunt necesare dacă $T_{\angle BBalt}$ se poate demonstra (printr-un contraexemplu) că este falsă
- implementarea se modifică și se reia demonstrarea parțial corectitudinii și apoi a terminării

- dacă pe baza teoremei terminării (vezi curs 07) toate condițiile de terminare $T_{\alpha_{ij}}$ sunt adevărate, atunci algoritmul se termină în raport cu precondiția $\varphi(X)$

- dacă algoritmul
  - este parțial corect: $V_{\alpha_{ij}}$ adevărate
  - se termină: $T_{\alpha_{ij}}$ adevărate

  $\Rightarrow$ algoritmul este total corect

---

2) Căutare binară

$\varphi(X): l[0] \le a \le l[n-1], l[0] \le l[1] \le \dots \le l[n-1]$

$\psi(X,Z): 0 \le p \le n-1, l[p] \le a < l[p+1]$

```
int search(List<Integer> l, int a) {
    • A: [φ(X): (l[0] ≤ a ≤ l[n-1]) ∧ (l[0] ≤ l[1] ≤ ... ≤ l[n-1])]
    int s = 0, d = l.size() - 1;
    while (s < d) {
        • B: [P_B(X,Y): l[s] ≤ a < l[d]]
        int m = (s+d)/2;
        if (a <= l.get(m))
            d = m;
        else s = m+1;
    }
    return s;
    • C: [ψ(X,Z): (0 ≤ p ≤ n-1) ∧ (l[p] ≤ a < l[p+1])]
}
```

I Parțial corectitudine

$X=(l_i, i=\overline{0,n-1}; a)$

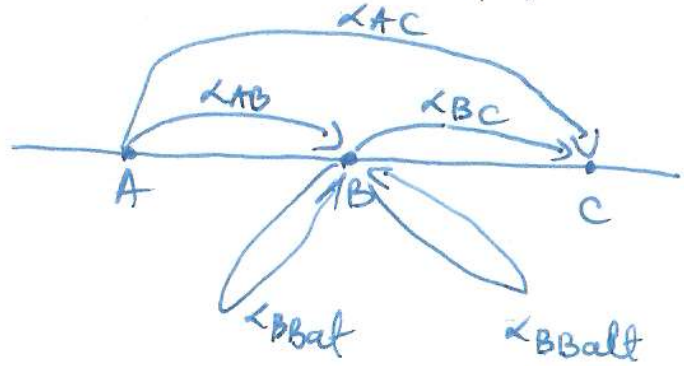
$Z=(s)$, $Y=(s, d, m)$

1) puncte de tăietură: A, B, C

2) predicate invariante: $\varphi(X)$, $P_B(X,Y)$, $\psi(X,Z)$

3) drumurile α_{ij}

-5 drumuri



4) condițiile de parcurgere a drumurilor α_{ij}

$R_{\alpha AB}(X,Y) = s<d$

$R_{\alpha BBat}(X,Y) = (a \le l[m]) < (s<d)$

$R_{\alpha BBalt}(X,Y) = (a > l[m]) < (s<d)$

$R_{\alpha BC}(X,Y) = s \ge d$

dar, $s=0$, $d=n-1$ $\Big\} \Rightarrow s=d$

indecși

$R_{\alpha AC}(X,Y) = R_{\alpha BC}(X,Y) = s=d$

5) funcțiile de transformare pe drumurile α_{ij}

$r_{\alpha AB}(X,Y) = (s, d, m) = (0, n-1, -)$

$r_{\alpha BBat}(X,Y) = (s, d, m) = \left(s, m, \frac{s+d}{2}\right)$

$r_{\alpha BBalt}(X,Y) = (s, d, m) = \left(m, d, \frac{s+d}{2}\right)$

$$r_{\alpha BC}(X,Y) = (s, d, m) = \begin{cases} \left(s, m, \frac{s+d}{2}\right), & a \le l[m] \\ \left(m, d, \frac{s+d}{2}\right), & a > l[m] \end{cases}$$

$r_{LAC}(X,Y) = (s, d, m) = (0, n-1, -)$

6) se construiesc și se demonstrează condițiile de verificare

- dacă toate condițiile de verificare sunt adevărate $\Rightarrow$

$\Rightarrow$ algoritmul este parțial corect în raport cu $(\varphi(X), \psi(X,Z))$

II Terminare

- se repetă pașii $1 \to 5$

6) mulțimea convenabil aleasă (M) este $\mathbb{N}$

7) funcțiile u_i

$u_A(X,Y) = (n-1) - 0 + 1 = n$

$u_B(X,Y) = d - s$

$u_C(X,Y) = 0$

descrește

8) se construiesc și se demonstrează condițiile de terminare

- dacă toate condițiile de terminare sunt adevărate $\Rightarrow$

$\Rightarrow$ algoritmul se termină în raport cu $\varphi(X)$

III Total corectitudine

- dacă s-a demonstrat parțial corectitudinea și terminarea $\Rightarrow$

$\Rightarrow$ **algoritmul este total corect**