Надовезивање програма

22

Нека су дати програми P и Q.
Може се појавити потреба за надовезивањем (композицијом) програма, тј. да направимо програм који ће након израчунавања по програму P наставити са израчунавањем по програму Q.
Приликом надовезивања програма Q на програм P могу се јавити следећа два техничка проблема:
Први проблем:

- Нека је $P = (I_1, \ldots, I_p)$.
- Израчунавање по програму P се завршава када се у бројачу појави број $s > p$.
- Да би се наставило израчунавање по програму Q (од његове прве инструкције) мора бити $s = p + 1$.

Надовезивање програма

- Дакле, да бисмо могли надовезати програм Q на програм P, програм P мора бити тако написан да се зауставља само ако се у бројачу појави $p+1$.
- Другим речима, у свим инструкцијама прелаза $J(m,n,k)$ програма P мора бити $k \leq p+1$. За такве програме кажемо да су у стандардној форми.

Дефиниција. За програм $P = (I_1, \dots, I_p)$ кажемо да је у стандардној форми ако за сваку његову инструкцију прелаза $J(m,n,k)$ важи $k \leq p+1$.

Лема. Сваком програму P можемо придружити програм P^* у стандардној форми такав да

$$P(a_1, \dots, a_n) \downarrow a \quad \text{акко} \quad P^*(a_1, \dots, a_n) \downarrow a$$

за све природне бројеве $a_1, \dots, a_n, a$.

Стандардна форма програма

Доказ. Нека je $P = (I_1, \dots, I_p)$. Да бисмо добили P^* довољно je изменити инструкције на следећи начин:

- ако I_s није инструкција прелаза, тада $I_s^* = I_s$
- ако je $I_s = J(m, n, k)$, тада
$$I_s^* = \begin{cases} I_s, & k \leq p+1 \\ J(m, n, p+1), & k > p+1 \end{cases}.$$

Тада, програм $P^* = (I_1^*, \dots, I_p^*)$ je у стандардној форми и важи
$$P(a_1, \dots, a_n) \downarrow a \quad \text{акко} \quad P^*(a_1, \dots, a_n) \downarrow a.$$
Дакле, овај проблем се лако превазилази писањем сваког програма у стандардној форми.

Пример.

- Програм $P = (I_1, I_2, I_3, I_4)$,

  $$\begin{array}{ll} I_1: & J(3,2,5) \\ I_2: & S(1) \\ I_3: & S(3) \\ I_4: & J(1,1,1) \end{array}$$

  је у стандардној форми.

Пример.

x ?

Програм

$$\begin{array}{ll} I_1 & J(1,2,10) \\ I_2 & S(3) \\ I_3 & J(1,3,7) \\ I_4 & S(2) \\ I_5 & S(3) \\ I_6 & J(1,1,3) \\ I_7 & T(2,1) \end{array}$$

који израчунава вредност функције $x \dot{-} 1$, није у стандардној форми.
Да би био у стандардној форми треба I_1 преправити у $J(1,2,8)$.

Композиција програма

23

Други проблем који се може појавити приликом надовезивања (композиције) програма Q на програм $P = (I_1, \ldots, I_p)$:
Инструкција $J(m, n, k)$ програма Q упућује на скок на k-ту инструкцију програма Q, уколико је $r_m = r_n$.
Међутим, k-та инструкција програма Q ће постати $p + k$-та инструкција композиције програма PQ.
Зато сваку инструкцију прелаза $J(m, n, k)$ програма Q треба преправити у $J(m, n, p + k)$.

Композиција програма

Дефиниција. Нека су $P = (I_1, \ldots, I_p)$ и $Q = (I'_1, \ldots, I'_q)$ програми у сандардној форми. Програм $PQ = (I_1, \ldots, I_p, I_{p+1}, \ldots, I_{p+q})$ је композиција програма P и Q ако

$$I_{p+i} = \begin{cases} I'_i, & \text{ако } I'_i \text{ није инструкција прелаза} \\ J(m, n, p+k), & \text{ако } I'_i = J(m, n, k) \end{cases} \quad \text{за } i = 1, 2,$$

Програми P и Q су онда потпрограми програма PQ.

Дубина програма

Претпоставимо да желимо да напишемо програм Q који садржи као потпрограм дати програм P. Често је ради чувања неких података потребно наћи регистре на које неће утицати израчунавање по програму P. Зато уводимо појам дубине програма.

- Програм P је коначан низ инструкција, па се током његовог извршавања мења садржај у коначно много регистара. Тај број регистара који се користе и чији се садржај мења приликом извршавања програма P се зове дубина програма P и обележава са $\delta(P)$.
- За све $n > \delta(P)$ регистри R_n остају непромењени током израчунавања по програму P, па их можемо користити као меморијски простор .

Дубина програма

Дефиниција. Најмањи природни број $\delta(P) \in \mathbb{N}$ такав да за све $n > \delta(P)$ регистри R_n остају непромењени током израчунавања по програму P зове се дубина програма P.

Нека је P програм у стандардној форми који израчунава вредности $f(x_1, x_2, \ldots, x_n)$. Када користимо програм P као потпрограм неког ширег програма често се дешава следеће:

- бројеви $x_1, x_2, \ldots, x_n$ могу бити записани у регистрима $R_{i_1}, \ldots, R_{i_n}$, а не у регистрима $R_1, \ldots, R_n$ како захтева програм P, па ове податке треба пребацити у $R_1, \ldots, R_n$,
- излаз $f(x_1, x_2, \ldots, x_n)$ који је потребан за даље израчунавање можемо сместити у R_s, а не у R_1,
- регистри $R_{n+1}, \ldots, R_{\delta(P)}$ могу садржавати неке податке, па их претходно треба испразнити, како би се омогућило израчунавање по програму P.

Дакле, програм P треба модификовати на следећи начин:

К 23

преместити $x_1, \ldots, x_n$ из регистара $R_{i_1}, \ldots, R_{i_n}$ у регистре $R_1, \ldots, R_n$

очистити регистре $R_{n+1}, \ldots, R_{\delta(P)}$

извршити програм P

пребацити излаз из R_1 у R_s

$$
\begin{array}{l}
\vdots \\
T(i_1, 1) \\
\vdots \\
T(i_n, n) \\
Z(n+1) \\
\vdots \\
Z(\delta(P)) \\
P \\
T(1, s) \\
\vdots
\end{array}
$$

Овај програм означавамо са $P[i_1, \ldots, i_n \to s]$. Он израчунава $f(r_{i_1}, \ldots, r_{i_n})$ и добијени резултат памти у R_s.

Супституција

24

Уобичајен начин за изградњу нових функција од постојећих је замена једних функција у другим.

Нека

$g_1 : D_1 \to \mathbb{N}, \quad g_2 : D_2 \to \mathbb{N}, \ldots, g_k : D_k \to \mathbb{N}$, где

$D_1, D_2, \ldots, D_k \subseteq \mathbb{N}^n$,

$h : D' \to \mathbb{N}$, $D' \subseteq \mathbb{N}^k$ и

$D = D_1 \cap \cdots \cap D_k \cap$

$\{(x_1, \ldots, x_n) | (g_1(x_1, \ldots, x_n), \ldots, g_k(x_1, \ldots, x_n)) \in D'\}$.

Дефиниција. За функција $f : D \to \mathbb{N}$ дефинисану са

$$f(x_1, x_2, \ldots, x_n) \stackrel{\text{def}}{=} h(g_1(x_1, x_2, \ldots, x_n), \ldots, g_k(x_1, x_2, \ldots, x_n)),$$

за $(x_1, x_2, \ldots, x_n) \in D$, кажемо да је супституција функција $g_1, g_2, \ldots, g_k$ у функцији h и пишемо $f = Sub(h; g_1, \ldots, g_k)$.

Напомена. Ако су све функције $g_1, \ldots, g_k, h$ тоталне, онда је и $f = Sub(h; g_1, \ldots, g_k)$ тотална функција.

Пример. Ако је

$g_1 : \mathbb{N} \to \mathbb{N}, \quad g_1(x) = x + 3,$

$g_2 : \mathbb{N} \to \mathbb{N}, \quad g_2(x) = 2x,$

$h : \mathbb{N}^2 \to \mathbb{N}, \quad h(x, y) = \min\{x, y\}$

онда је

$f : \mathbb{N} \to \mathbb{N},\ f(x) = h(g_1(x), g_2(x)) = \min\{x + 3, 2x\}$

(супституција g_1 и g_2 у h, тј. $f = Sub(h; g_1, g_2)$)

$f_1 : \mathbb{N}^2 \to \mathbb{N},\ f_1(x, y) = g_1(h(x, y)) = \min\{x, y\} + 3$ (супституција h у g_1)

$f_2 : \mathbb{N}^2 \to \mathbb{N},\ f_2(x, y) = g_2(h(x, y)) = 2\min\{x, y\}$(супституција h у g_2)

Теорема. Ако су $g_1, g_2, \ldots, g_k$ и h израчунљиве функције, онда је израчунљива и супституција f функција $g_1, g_2, \ldots, g_k$ у функцији h.

Доказ.

- По претпоставци, функције $g_1, g_2, \ldots, g_k$ и h су израчунљиве, па постоје програми $G_1, G_2, \ldots, G_k$ и H у стандардној форми који израчунавају вредности, редом, функција $g_1, g_2, \ldots, g_k$ и h. Да бисмо доказали да је и функција f израчунљива, треба одредити програм F за израчунавање њених вредности.
- За задату почетну конфигурацију $(x_1, \ldots, x_n)$ најпре се извршавају програми $G_1, G_2, \ldots, G_k$ који израчунавају редом вредности
  $g_1(x_1, x_2, \ldots, x_n),\ g_2(x_1, x_2, \ldots, x_n), \ldots, g_k(x_1, x_2, \ldots, x_n)$.
  Те вредности су улаз за програм H који израчунава вредност
  $f(x_1, x_2, \ldots, x_n) =$
  $h(g_1(x_1, x_2, \ldots, x_n), \ldots, g_k(x_1, x_2, \ldots, x_n))$.

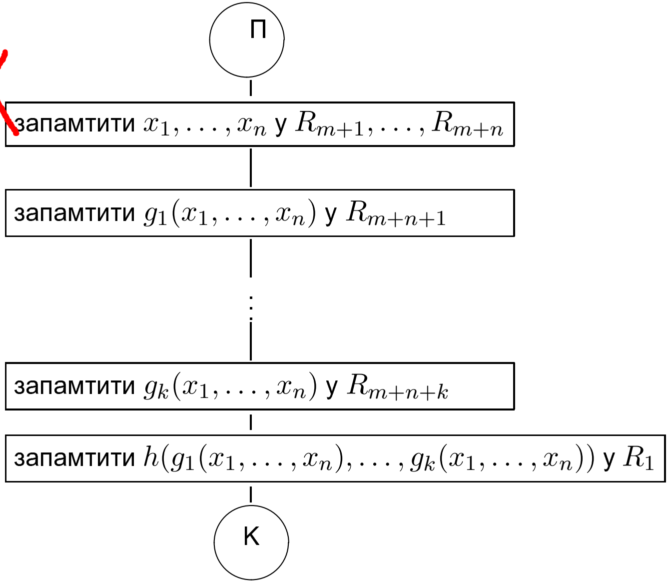
наставак доказа. При томе је потребно обезбедити довољно простора за израчунавања по програмима $G_1, G_2, \ldots, G_k$ и H и сачувати све потребне податке.
Нека је

$$m = \max\{n, k, \delta(G_1), \ldots, \delta(G_k), \delta(H)\}.$$

Бројеве $x_1, x_2, \ldots, x_n$ који су потребни при извршавању сваког од програма $G_1, G_2, \ldots, G_k$, чуваћемо у регистрима $R_{m+1}, \ldots, R_{m+n}$.
Садржај ових регистара неће се мењати ни при једном од извршавања програма $G_1, \ldots, G_k, H$.
Резултате израчунавања по програмима $G_1, G_2, \ldots, G_k$ за улаз $(x_1, x_2, \ldots, x_n)$, чуваћемо, редом, у регистрима $R_{m+n+1}, \ldots, R_{m+n+k}$.

R_1	$\ldots$	R_m	R_{m+1}	$\ldots$	R_{m+n}	R_{m+n+1}	$\ldots$	R_m
	$\ldots$		x_1	$\ldots$	x_n	$g_1(x_1, \ldots, x_n)$	$\ldots$	$g_k($

K 24


П
запамтити $x_1, \ldots, x_n$ у $R_{m+1}, \ldots, R_{m+n}$
запамтити $g_1(x_1, \ldots, x_n)$ у R_{m+n+1}
запамтити $g_k(x_1, \ldots, x_n)$ у R_{m+n+k}
запамтити $h(g_1(x_1, \ldots, x_n), \ldots, g_k(x_1, \ldots, x_n))$ у R_1
К