Γλώσσες Προγραμματισμού ΙΙ

Άσκηση 10:
Αξιωματική σημασιολογία

Δανάη Ευσταθίου, 10ο εξάμηνο

ΑΜ : 03115122

Περιγραφή

Για την άσκηση αυτή ζητείται να αποδειχθεί η ορθότητα μιας δοσμένης συνάρτησης. Συγκεκριμένα, δίνεται συνάρτηση που υπολογίζει την μέγιστη απόσταση μεταξύ δύο εμφανίσεων του ίδιου αριθμού στην ακολουθία. Για να αποδειχθεί η ορθότητά της, επιλέχθηκε ο πρώτος τρόπος που αναφέρεται στην εκφώνηση, δηλαδή η χρήση απλής προστακτικής γλώσσας.

Απόδειξη ορθότητας

Αρχικά, γράφουμε τη δοσμένη συνάρτηση σε απλή προστακτική γλώσσα και προσθέτουμε λογικές συνθήκες σε τμήματα της συνάρτησης, οι οποίες θα μας βοηθήσουν να αποδείξουμε την τελευταία λογική συνθήκη, η οποία και αποτελεί την συνθήκη ορθότητας της συνάρτησης.

$\{\}$

i := 0

$\{i = 0\}$

while $i <= MAXV$ **do**

$\quad p[i] := -1$

$\{\forall i : 0 \leq i \leq MAXV \Rightarrow p[i] = -1\}$ (1)

$i := 0$

$r := 0$

$\{r = 0 \ \wedge \ i = 0 \ \wedge \ \forall j : 0 \leq j \leq MAXV \Rightarrow p[j] = -1\}$ (2)

$skip$

$\{r \neq 0 \Leftrightarrow (\exists j, k : x[j] = x[k] \ \wedge \ j < k < 0 \ \wedge \ r = k - j) \ \wedge \ ((\forall j', k' : (j' \neq j \ \vee \ k' \neq k) \ \wedge \ x[j'] = x[k'] \ \wedge \ j' < k' < 0 \Rightarrow r \geq k' - j'))\}$ (3)

while $i < N$ **do**

$\quad \{r \neq 0 \Leftrightarrow (\exists j, k : x[j] = x[k] \ \wedge \ j < k < i < N \ \wedge \ r = k - j \ \wedge \ ((\forall j', k' : (j' \neq j \ \vee \ k' \neq k) \ \wedge \ x[j'] = x[k'] \ \wedge \ j' < k' < i < N) \Rightarrow r \geq k' - j'))\}$ (4)

$\quad$ **if** $p[x[i]] = -1$ **then**

$\quad\quad p[x[i]] := i$

$\quad$ **else if** $i - p[x[i]] > r$ **then**

$\quad\quad r := i - p[x[i]]$

$\quad i := i + 1$

$\quad \{r \neq 0 \Leftrightarrow (\exists j, k : x[j] = x[k] \ \wedge \ j < k < i \leq N \ \wedge \ r = k - j \ \wedge \ ((\forall j', k' : (j' \neq j \ \vee \ k' \neq k) \ \wedge \ x[j'] = x[k'] \ \wedge \ j' < k' < i \leq N) \Rightarrow r \geq k' - j'))\}$ (5)

$\{r \neq 0 \Leftrightarrow (\exists j, k : x[j] = x[k] \ \wedge \ j < k < N \ \wedge \ r = k - j \ \wedge \ ((\forall j', k' : (j' \neq j \ \vee \ k' \neq k) \ \wedge \ x[j'] = x[k'] \ \wedge \ j' < k' < N) \Rightarrow r \geq k' - j'))\}$ (6)

Για να μπορέσουμε να αποδείξουμε την ορθότητα του αλγορίθμου, και συγκεκριμένα την ορθότητα του δεύτερου while, θα χρησιμοποιήσουμε επαγωγή ως προς το πλήθος των επαναλήψεων i. Από την συνθήκη (2) η συνθήκη (3) ισχύει, αφού αφορά στην τετριμμένη περίπτωση $i = 0$. Συγκεκριμένα το r δεν είναι διάφορο του 0 από την (2) και ακόμα δεν ισχύει ότι $\exists j, k : j < k < 0$, επομένως $false \Leftrightarrow false$ είναι αληθές. Η συνθήκη (3) θα είναι η επαγωγική μας βάση.

Ως επαγωγική υπόθεση θεωρούμε την συνθήκη (4) και θα πρέπει να αποδείξουμε ότι ισχύει η συνθήκη (5) εφαρμόζοντας το επαγωγικό βήμα στην (4). Αν στο σημείο της συνθήκης (4) στο r είναι αποθηκευμένη η μέγιστη απόσταση μεταξύ δύο εμφανίσεων του ίδιου αριθμού στην ακολουθία x μέχρι το i εκείνη τη στιγμή, τότε το if μπορεί να έχει τα εξής αποτελέσματα:

1. Αν ο $x[i]$ εμφανίζεται ως αριθμός για πρώτη φορά στην ακολουθία, δηλαδή το $p[x[i]] = -1$, τότε αποθηκεύεται η θέση του στον πίνακα p, δηλαδή $p[x[i]] = i$.
2. Διαφορετικά, δηλαδή αν έχει ξαναεμφανιστεί ο αριθμός $x[i]$, η απόστασή του από την πρώτη εμφάνισή του είναι υποψήφια μέγιστη απόσταση. Αν είναι όντως μεγαλύτερη από την μέχρι τώρα υπολογισμένη μέγιστη απόσταση r, τότε γίνεται αυτή η μέγιστη απόσταση.

Και για τις δύο συνθήκες (4), (5) η περίπτωση $r = 0$ είναι τετριμμένη και τις επαληθεύει πάντα, αφού $\nexists j, k : x[j] = x[k] \;\wedge\; j < k < N \;\wedge\; r = k - j$ και διαφορετικά $r \neq 0$. Εξετάζουμε λοιπόν παρακάτω μόνο την περίπτωση $r \neq 0$. Επομένως, μετά την εκτέλεση του if διακρίνουμε δύο πιθανές περιπτώσεις:

1. Είτε δεν είχε ξαναεμφανιστεί ο αριθμός $x[i]$, επομένως δεν επηρεάζεται ο αριθμός r και η συνθήκη (5) ισχύει. Συγκεκριμένα πρόκειται για τα ίδια j, k για τα οποία $x[j] = x[k] \;\wedge\; j < k < i < N \;\wedge\; r = k - j$ και στις δύο συνθήκες (4), (5) και αφού ο $x[i]$ δεν έχει ξαναεμφανιστεί δεν υπάρχει περίπτωση στην συνθήκη (5) για $k' = i$, που είναι η μόνη έξτρα περίπτωση σε σχέση με την συνθήκη (4), να ισχύει ότι $x[j'] = x[k']$.
2. Είτε έχει ξαναεμφανιστεί ο αριθμός $x[i]$ και:
   (a) Η απόστασή του από την πρώτη εμφάνισή του δεν είναι μεγαλύτερη του r, οπότε πρόκειται και πάλι για τα ίδια j, k για τα οποία $x[j] = x[k] \;\wedge\; j < k < i < N \;\wedge\; r = k - j$ και στις δύο συνθήκες (4), (5) και ισχύει ότι $(\forall j' : x[j'] = x[i] \;\wedge\; j' < i < N) \Rightarrow r \geq k' - j')$, αφού δεν εκτελέστηκε η εντολή μέσα στο else if και δεν άλλαξε το r.
   (b) Η απόστασή του από την πρώτη εμφάνισή του είναι μεγαλύτερη του r, επομένως λόγω του else if $r = i - p[x[i]]$, $r \neq 0$, $\exists j : x[j] = x[i] \;\wedge\; j < i < N \;\wedge\; r = i - j$, όπου συγκεκριμένα j η πρώτη θέση εμφάνισης του $x[i]$ και $(\forall j', k' : (j' \neq j \;\vee\; k' \neq i) \;\wedge\; x[j'] = x[k'] \;\wedge\; j' < k' < N) \Rightarrow r \geq k' - j'$, αφού η τελευταία πρόταση ισχύει για όλα τα j', k' της συνθήκης (4), καθώς και για την υπολειπόμενη περίπτωση των j, k της συνθήκης (4), μιας και η τιμή του r της συνθήκης (4) αντικαταστάθηκε από τη νέα μέγιστη απόσταση στη συνθήκη (5).

Με αυτόν τον τρόπο δείξαμε την ορθότητα της συνάρτησης, και συγκεκριμένα ότι η συνθήκη ορθότητας, η οποία σημαίνει ότι η συνάρτηση βρίσκει την μέγιστη απόσταση μεταξύ δύο εμφανίσεων του ίδιου αριθμού στην ακολουθία x μήκους N, ισχύει για κάθε πιθανή εκτέλεση. Ακόμα, η συνάρτηση αυτή τερματίζει, αφού το i θα φτάσει σίγουρα κάποια στιγμή την τιμή N και το while loop θα τερματίσει.