

Γλώσσες Προγραμματισμού II

Άσκηση 7:

Συστήματα τύπων - μηχανή στοίβας

Δανάη Ευσταθίου, 10ο εξάμηνο

AM : 03115122

Περιγραφή

Για την άσκηση αυτή ζητείται ο ορισμός ενός συστήματος τύπων για μια γλώσσα μηχανής στοίβας. Για τον σκοπό αυτό θεωρήθηκαν 3 βασικοί τύποι:

1. ο `Int` που αντιστοιχεί στους ακέραιους,
2. ο `Bool` που αντιστοιχεί στις λογικές τιμές `true`, `false` και
3. ο `Unit` που αντιστοιχεί στον κενό τύπο.

Αρχικά, θα πρέπει να βρεθεί ένας τρόπος αναπαράστασης των τύπων των στοιχείων της στοίβας. Καθώς δεν μπορούν να είναι γνωστά στο στάδιο του `type checking` όλα τα στοιχεία της στοίβας, αφού αυτό θα ισοδυναμούσε με πλήρη εκτέλεση του προγράμματος, πρέπει να επιλεγεί ένα μέγιστο μέγεθος στοιχείων της στοίβας, για τα οποία θα βρεθούν οι τύποι τους. Στην συγκεκριμένη άσκηση επιλέχθηκε να σημειώνεται πλειάδα τεσσάρων στοιχείων, που είναι ο μικρότερος αριθμός στοιχείων που επιτρέπει το πρόγραμμα που δίνεται ως παράδειγμα να περάσει τον έλεγχο τύπων.

Η πλειάδα αυτή θα αντιστοιχεί στους τύπους των 4 στοιχείων της κορυφής της στοίβας, με δεξιότερο τύπο αυτόν που αντιστοιχεί στην κορυφή της στοίβας. Όταν εισάγεται επομένως ένα στοιχείο στην στοίβα, θα εισάγεται ο τύπος του από τα δεξιά της πλειάδας, δηλαδή αν η πλειάδα αρχικά είναι της μορφής $\langle t_1, t_2, t_3, t_4 \rangle$, μετά την είσοδο του νέου στοιχείου θα είναι της μορφής $\langle t_2, t_3, t_4, t_5 \rangle$, όπου t_5 ο τύπος του νέου στοιχείου. Από την άλλη, όταν αφαιρείται ένα στοιχείο από την στοίβα, θα εξέρχεται ο τύπος του από τα δεξιά της πλειάδας και θα εισέρχεται ο τύπος `Unit` από τα αριστερά, αφού έχουμε 'μνήμη' μόνο των τεσσάρων κορυφαίων στοιχείων, δηλαδή αν η πλειάδα αρχικά είναι της μορφής $\langle t_1, t_2, t_3, t_4 \rangle$, μετά την έξοδο του κορυφαίου στοιχείου θα είναι της μορφής $\langle Unit, t_1, t_2, t_3 \rangle$. Πριν την έναρξη του προγράμματος η πλειάδα αυτή είναι αρχικοποιημένη σε $\langle Unit, Unit, Unit, Unit \rangle$.

Είναι κατανοητό πως αυτός ο τρόπος αναπαράστασης του τύπου της στοίβας θα δώσει σφάλμα τύπου σε προγράμματα που θα μπορούσαν να τρέξουν χωρίς να κολλήσουν, αλλά όσα προγράμματα περάσουν τον έλεγχο τύπων είναι σίγουρο ότι δεν θα κολλήσουν.

Κανόνες Τύπων

Για το σύστημα τύπων αναπτύχθηκαν οι παρακάτω κανόνες τύπων, όσον αφορά την επίδραση των εντολών τον τύπο της πλειάδας μήκους 4, όπως αυτή ορίστηκε παραπάνω. Οι κανόνες είναι είτε της μορφής:

$$x : \langle a, b, c, d \rangle \rightarrow \langle e, f, g, h \rangle,$$

που σημαίνει ότι η εντολή x λαμβάνει είσοδο με τύπο $\langle a, b, c, d \rangle$ και δίνει έξοδο με τύπο $\langle e, f, g, h \rangle$,

είτε της:

$$\frac{x, y}{z}$$

που σημαίνει ότι αν ισχύουν τα x, y τότε ισχύει το z .

Για την αναπαράσταση οποιουδήποτε εκ των τριών τύπων *Int*, *Bool*, *Unit* χρησιμοποιούνται οι χαρακτήρες του αγγλικού αλφαβήτου (a, b, c, \dots), ενώ για την αναπαράσταση των τύπων *Int*, *Bool* χρησιμοποιούνται οι χαρακτήρες του ελληνικού αλφαβήτου ($\alpha, \beta, \gamma, \dots$).

Με αυτόν τον τρόπο, οι κανόνες διαμορφώθηκαν ως εξής:

$$\begin{aligned} n &: \langle a, b, c, d \rangle \rightarrow \langle b, c, d, Int \rangle \\ true, false &: \langle a, b, c, d \rangle \rightarrow \langle b, c, d, Bool \rangle \\ +, * &: \langle a, b, Int, Int \rangle \rightarrow \langle Unit, a, b, Int \rangle \\ - &: \langle a, b, c, Int \rangle \rightarrow \langle a, b, c, Int \rangle \\ / &: \langle a, b, Int, Int \rangle \rightarrow \langle a, b, Int, Int \rangle \\ <: &: \langle a, b, Int, Int \rangle \rightarrow \langle Unit, a, b, Bool \rangle \\ =: &: \langle a, b, \alpha, \alpha \rangle \rightarrow \langle Unit, a, b, Bool \rangle \\ and &: \langle a, b, Bool, Bool \rangle \rightarrow \langle Unit, a, b, Bool \rangle \\ not &: \langle a, b, c, Bool \rangle \rightarrow \langle a, b, c, Bool \rangle \\ dup &: \langle a, b, c, \alpha \rangle \rightarrow \langle b, c, \alpha, \alpha \rangle \\ pop &: \langle a, b, c, \alpha \rangle \rightarrow \langle Unit, a, b, c \rangle \\ swap &: \langle a, b, \alpha, \beta \rangle \rightarrow \langle a, b, \beta, \alpha \rangle \\ swap2 &: \langle a, \alpha, \beta, \gamma \rangle \rightarrow \langle a, \gamma, \beta, \alpha \rangle \\ nop &: \langle a, b, c, d \rangle \rightarrow \langle a, b, c, d \rangle \\ \\ p_1 &: \langle Unit, a, b, c \rangle \rightarrow \langle d, e, f, g \rangle, \quad p_2 : \langle Unit, a, b, c \rangle \rightarrow \langle d, e, f, g \rangle \\ \\ \hline cond[p_1|p_2] &: \langle a, b, c, Bool \rangle \rightarrow \langle d, e, f, g \rangle \\ \\ p &: \langle Unit, a, b, c \rangle \rightarrow \langle Unit, a, b, c \rangle \\ \\ \hline loop[p] &: \langle a, b, c, Bool \rangle \rightarrow \langle Unit, a, b, c \rangle \\ \\ p_1 &: \langle a, b, c, d \rangle \rightarrow \langle e, f, g, h \rangle, \quad p_2 : \langle e, f, g, h \rangle \rightarrow \langle i, j, k, l \rangle \\ \\ \hline p_1 \ p_2 &: \langle a, b, c, d \rangle \rightarrow \langle i, j, k, l \rangle \end{aligned}$$

Καθώς για τους υπόλοιπους κανόνες είναι σχετικά προφανής η ιδέα για την διαμόρφωσή τους, θα ήταν σκόπιμο να γίνει διευκρίνιση του γιατί έγιναν οι συγκεκριμένες σχεδιαστικές επιλογές για τους κανόνες των *cond* και *loop*:

- Ο κανόνας για την *cond* διαμορφώθηκε έτσι, ώστε τα δύο προγράμματα του *cond* να παίρνουν μια πλειάδα με τον τύπο *Unit* στην πρώτη θέση. Η επιλογή αυτή έγινε, διότι θεωρήσαμε ότι για να εκτελεστεί το *cond* θα πρέπει να υπάρχει μια τιμή *Bool* στην κορυφή της στοίβας, η οποία με την έξοδό της από αυτή θα δώσει την τιμή *Unit* στην πρώτη θέση της πλειάδας. Ακόμα, θεωρήσαμε ότι τα προγράμματα p_1, p_2 δίνουν τον ίδιο τύπο στην έξοδο, προκειμένου και ο τύπος εξόδου της *cond* να είναι απλά ο ίδιος με αυτόν των προγραμμάτων p_1, p_2 .
- Η εντολή *loop* παίρνει ως όρισμα ένα πρόγραμμα p , το οποίο δίνει έξοδο του ίδιου τύπου με την είσοδο. Η επιλογή αυτή έγινε προκειμένου οι πιθανές συνεχόμενες εκτελέσεις του p , που ισοδυναμούν με πιθανή παράθεση του p με τον εαυτό του πολλαπλές φορές, να μην δημιουργούν σύγκρουση με τον κανόνα τύπων για την παράθεση. Και σε αυτή την περίπτωση ισχύει το ίδιο με την περίπτωση του *cond* για την τιμή *Unit* στην πρώτη θέση της πλειάδας του τύπου εισόδου του p .