Σημασιολογική Ανάλυση

Μεταγλωττιστές,

Χειμερινό εξάμηνο 2018-2019

Σημασιολογικός Αναλυτής

- Σημασιολογική Ανάλυση:
  - Η διαδικασία εκτέλεσης σημασιολογικών ελέγχων.
    - έλεγχος τύπου

Code:

```
float a = "example";
```

Σφάλμα σημασιολογικού ελέγχου:

error: incompatible types in initialization

Σημασιολογική Ανάλυση

- Το πρόγραμμα είναι λεκτικά καλώς διαμορφωμένο:
  - Τα αναγνωριστικά έχουν έγκυρα ονόματα.
  - Τα stings τερματίζονται σωστά.
  - …
- Το πρόγραμμα είναι συντακτικά καλώς διαμορφωμένο:
  - Οι δηλώσεις έχουν τη σωστή δομή.
  - Οι εκφράσεις είναι συντακτικά έγκυρες.
  - …
- Αυτό σημαίνει ότι το πρόγραμμα είναι σωστό;

Σημασιολογική Ανάλυση

- Βεβαιωθείτε ότι το πρόγραμμα έχει μια καλά καθορισμένη έννοια.
- Βεβαιωθείτε για τις ιδιότητες του προγράμματος που δεν εμπίπτουν στις πρώτες φάσεις:
  - Οι μεταβλητές δηλώνονται πριν χρησιμοποιηθούν.
  - Οι εκφράσεις έχουν τους σωστούς τύπους.
  - ...
- Μόλις ολοκληρώσουμε τη σημασιολογική ανάλυση, ξέρουμε ότι το πρόγραμμα εισόδου του χρήστη είναι νόμιμο.

Σύνταξη και Σημασιολογία

- Σε οποιαδήποτε γλώσσα (προγραμματισμού ή μη) υπάρχουν δύο βασικές έννοιες:
  - Σύνταξη: μορφή και δομή καλώς σχηματισμένων ακολουθιών συμβόλων (προγραμμάτων).
  - Σημασιολογία: αφορά την ερμηνεία των ορθά σχηματισμένων ακολουθιών (προγραμμάτων).
    - Στατική Σημασιολογία.
    - Δυναμική Σημασιολογία.

Στατική Σημασιολογία

- Έλεγχος πέρα από τον συντακτικό έλεγχο για την ορθότητα ενός προγράμματος για τον εντοπισμό σημασιολογικών σφαλμάτων.
  - Δήλωση μεταβλητής μια μόνο φορά σε μια εμβέλεια.
  - Ορθή χρήση τελεστών σε σχέση με τον τύπο των ορισμάτων.
  - Ύπαρξη μιας εντολής break εκτός κατάλληλου βρόγχου.

Δυναμική Σημασιολογία

- Απόδοση ερμηνείας σε προγράμματα που περιγράφει την συμπεριφορά τους.
- Τρόποι περιγραφής:
  - Λειτουργική Σημασιολογία (operational semantics): περιγράφει συμπεριφορά σαν υπολογιστικά βήματα (π.χ., εντολή ανάθεσης).
  - Δηλωτική Σημασιολογία (denotational semantics): περιγραφή με κάποιοα μαθηματική συνάρτηση (από το πεδίο των δεδομένων εισόδου στο πεδίο των αποτελεσμάτων).
  - Αξιωματική Σημασιολογία (axiomatic semantics): η ερμηνεία καθορίζεται έμμεσα μέσω λογικών προτάσεων που περιγράφουν ιδιότητες του προγράμματος.

Έλεγχοι που διεξάγονται:

- Έλεγχοι τύπων:
  - τύποι μεταβλητών, συστήματα τύπων,κλπ.
- Έλεγχοι ύπαρξης ονομάτων:
  - η χρήση μιας μεταβλητής επιτρέπεται μετά τη δήλωση της (πίνακας συμβόλων).
- Έλεγχοι μοναδικότητας:
  - Δύο ονόματα ετικετών, συναρτήσεων, κλπ δεν μπορούν να έχουν το ίδιο όνομα (πίνακας συμβόλων).
- Ελεγχοι συνέπειας:
  - σε εντολές που ξεκινούν και τελειώνουν με το όνομα της δηλωθείσας δομής
- Έλεγχοι ροής:
  - Όχι continue έξω από βρόχο.
  - Οι μεταβλητές αρχικοποιούνται πριν τη χρήση τους.

Υλοποίηση Σημασιολογικής Ανάλυσης

- Μεταγλωττιστής πολλαπλών περασμάτων:
  - Δημιουργία συντακτικού δένδρου και έπειτα έλεγχος.
- Μεταγλωττιστής απλού περάσματος:
  - Η ανάλυση γίνεται σταδιακά παράλληλα με την συντακτική ανάλυση.
- Θεωρητικά υπάρχει η ανάγκη για υποστήριξη όλων των παραπάνω από κάποιου είδους γραμματικής.

Συστήματα Τύπων

- Σύστημα τύπων (type system): διαχείριση και έλεγχος τύπων.
- Βασικοί τύποι (basic types): πρωταρχικοί τύποι της γλώσσας.
  - integer, boolean, real, char.
- Σύνθετοι τύποι (composite types): τύποι που προκύπτουν από "κατασκευαστές τύπων" με συνδυασμό άλλων τύπων:
    - εγγραφές, δείκτες, συναρτήσεις…

Κατηγορήματα

- Κάθε σύμβολο στο Bison συνοδεύεται και από μία τιμή.
- Δήλωση του τύπου του κατηγορήματος με χρήση **%type** για μη τερματικά σύμβολα.
  - Και τα τερματικά σύμβολα έχουν τιμές.
- Οι τιμές των συντιθέμενων κατηγορημάτων N συμβόλων του RHS προσπελαύνονται μέσω των $1-$N.
  - Ένα κατηγόρημα για κάθε σύμβολο.
- Τα κατηγορήματα του LHS προσπελαύνονται μέσω του $$.

Παραδείγματα

- Υπάρχει δυνατότητα για πρόσβαση και στα κληρονομούμενα κατηγορήματα.
  - Πρόσβαση μέσω των $0,$-1….
- Παράδειγμα:

```
Declaration : class type nanelist;
 class: GLOBAL {$$=1;}
 | LOCAL {$$=2;}
 ;
 type: REAL {$$=1;}
 | INTEGER {$$=2;}
 ;
 namelist : NAME {mksymbol($0,$-1,$1);}
 | namelist NAME {mksymbol($0,$-1,$2);}
 ;
```

Σημασιολογία

- Σε κάθε σύμβολο της γραμματικής μπορεί να αποδοθεί μια σημασιολογική τιμή.
- Ο τύπος των σημασιολογικών τιμών καθορίζεται από το macro YYSTYPE.
  - #define YYSTYPE double
- Οι σημασιολογικές τιμές μπορούν να είναι διαφορετικού τύπου για διαφορετικά σύμβολα. Οι τύποι αυτοί δε δηλώνονται με ορισμό του YYSTYPE, αλλά στο πρώτο μέρος ως εξής:

```
%{ typedef enum {TY_int, TY_real, TY_boolean} Type;
        %union {
                    char * n;
                    Type t;
                    struct {
                         Type type;
                        /* other fields */
                    } v; }
```

Σημασιολογία

- Με $$ συμβολίζεται η σημασιολογική τιμή του αριστερού μέλους ενός κανόνα.
- Τα σύμβολα $n, όπου n>0, παριστάνουν τη σημασιολογική τιμή του n-οστού συμβόλου του δεξιού μέλους του κανόνα. Το πρώτο σύμβολο αντιστοιχεί στο $1, κ.ο.κ.
- Οι σημασιολογικές τιμές των τερματικών συμβόλων συνήθως καθορίζονται από το λεκτικό αναλυτή και τοποθετούνται στη μεταβλητή yylval.

Έλεγχος Τύπων

- Στο σχηματισμό των εκφράσεων αποδίδονται σημασιολογικές τιμές σύμφωνα με τα (μη-)τερματικά σύμβολα που τις παράγουν:
- Στην περίπτωση των σταθερών, παίρνουμε τον τύπο, όπως αυτός έχει αναγνωρισθεί από τον λεκτικό αναλυτή.
- Στην περίπτωση των μεταβλητών, ο τύπος τους αποθηκεύεται στον πίνακα συμβόλων κατά τον ορισμό τους.
- Στην περίπτωση εκφράσεων με τελεστές, υπάρχουν περιπτώσεις που επιτρέπονται μόνο συγκεκριμένοι τύποι.

Έλεγχος Τύπων

- Παραδείγματα:
  - Ο τελεστής ακεραίου υπολοίπου mod (ή %) δέχεται μόνο τελούμενα ακεραίου τύπου

```
expr: expr “%” expr
        { if ($1.type==TY_INT && $3.type==TY_INT)
                $$.type=TY_INT;
         else
                yyerror(“type mismatch”);}
```

  - Στον κανόνα περιγραφής του while η συνθήκη πρέπει να έχει τύπο boolean

```
stmt: “while” condition while_body
        { if ($2.type != TY_boolean)
                yyerror(“condition type mismatch”);}
```

Μετατροπές Τύπων

- Όταν επιτρέπεται ρητή μετατροπή τύπου (type casting) ανάμεσα σε συμβατούς τύπους.

```
expr : ‘(‘ typename ‘)’ expr  {
                if (isCastAllowed ($2, $4.type))
                        $$.type = $2;
                else
                        yyerror(“illegal type cast”); }
```

- Επιτρέπεται επίσης η έμμεση μετατροπή τύπου (type coersion) όπου μια έκφραση τύπου t χρησιμοποιείται σε ένα σημείο όπου αναμενόταν μια έκφραση τύπου t’, τότε η έκφραση μετατρέπεται αυτόματα στον τύπο t’ (άν αυτό είναι δυνατό)

Υπερφόρτωση Τελεστών

- Όταν ένας τελεστής μπορεί να εφαρμοστεί σε τελούμενα διαφορετικών τύπων υλοποιώντας διαφορετικές πράξεις, λέμε ότι έχουμε υπερφόρτωση (overloading) του τελεστή

```
expr: expr ‘+’ expr   {
                if ($1.type == TY_int)
                        if($3.type == TY_int)       $$.type = TY_int;
                        else if($3.type == TY_real) $$.type = TY_real;
                        else       yyerror(“type mismatch”);
                else if($1.type == TY_real)
                         if($3.type == TY_int)      $$.type = TY_real;
                        else if($3.type == TY_real) $$.type = TY_real;
                        else       yyerror(“type mismatch”);
                else       yyerror(“type mismatch”);
```

Παραδείγματα:

```
%{typedef enum { TY_int, TY_real, TY_bool }
Type;%}
%union{char * n;
Type t;
struct { Type type;
/* other fields */ } v;
}
%type<n> T_id
%type<t> typename
%type<v> expression
```

```
expression : T_intconst { $$.type = TY_int; }
           | T_realconst { $$.type = TY_real;}
           | ‘(‘expression ‘)’{ $$.type = $2.type; } ;
```

- Expression : T_id
- Έλεγχοι:
  - έχει δηλωθεί το T_id στον πίνακα συμβόλων?
  - τι τύπο έχει?

Παραδείγματα:

```
expression : T_id
{
    Entry * id = lookup($1);
    if (id != NULL && id->kind == K_variable)
                $$.type = id->type;
   else
                yyerror("identifier not found");
} ;
```

- Έλεγχος με την lookup εάν υπάρχει στον πίνακα συμβόλων.
- Ένθεση στο πεδίο type το type που βρέθηκε

Για την εργασία

- Να αποτιμά τα κατηγορήματα των συμβόλων της γραμματικής.
- Να προσθέτει στον ΠΣ πληροφορίες για τα αναγνωριστικά που έχουν εισαχθεί από το ΛΑ ή το ΣΑ:
  - Να κωδικοποιούνται στον ΠΣ οι τύποι και τα συνώνυμα τύπων.
  - Να προσδιορίζονται οι τύποι των μεταβλητών και των συναρτήσεων κατά τη δήλωσή τους.
  - Να αποδίδονται τιμές σε ονόματα σταθερών ή πιθανές αρχικές τιμές σε ονόματα μεταβλητών.
  - Να προσδιορίζονται ο αριθμός, οι τύποι, ο τρόπος περάσματος και τα ονόματα των παραμέτρων των υποπρογραμμάτων που δηλώνονται στο πρόγραμμα

ΕΡΩΤΗΣΕΙΣ/ΑΠΟΡΙΕΣ