

Εργασία στην Σχεδίαση Γλωσσών Προγραμματισμού

Δημήτριος Γκουντέλος, 3663

16 Ιουνίου 2021

Περιγραφή

Από την εργασία υλοποιήθηκε η **λεξική, συντακτική και σημασιολογική** ανάλυση με **πίνακα συμβόλων** και εκτύπωση του συντακτικού δέντρου και όχι ολόκληρη, δηλαδή χωρίς την παραγωγή κώδικα στη γλώσσα στόχο. Το εργαλείο που χρησιμοποιήθηκε ήταν το Antlr4 στο περιβάλλον IntelliJ IDEA. Η λεξική και η συντακτική ανάλυση γίνονται στο αρχείο "SimpleFortran.g4", που βρίσκεται στον φάκελο "src" (γίνονται και τα 2 στο ίδιο αρχείο). Η σημασιολογική ανάλυση με τον πίνακα συμβόλων και η εκτύπωση του συντακτικού δέντρου γίνονται στα αρχεία "Runner.java" και "Visitor.java", που βρίσκονται επίσης στον φάκελο "src".

Λεξική ανάλυση

Στη λεξική ανάλυση, αρχικά ορίζονται οι δεσμευμένες λέξεις της γλώσσας. Αυτές είναι:

- function
- subroutine
- end
- common
- integer
- real
- complex
- logical
- character
- string
- list

- data
- continue
- goto
- call
- read
- write
- length
- new
- if
- then
- else
- endif
- do
- enddo
- stop
- return

Για να μην υπάρχει διάκριση πεζών και κεφαλαίων γραμμάτων, όλες οι δεσμευμένες λέξεις ορίζονται με την μορφή: T_FUNCTION : F U N C T I O N, T_END : E N D κλπ και το κάθε γράμμα έχει τη μορφή: F : 'F' | 'f', E : 'E' | 'e' κλπ.

Τα αναγνωριστικά της γλώσσας (ID) ξεκινούν με αλφαβητικό χαρακτήρα ακολουθούμενο από μηδέν ή περισσότερους αλφαριθμητικούς χαρακτήρες. Μετά τον υποχρεωτικό αλφαβητικό χαρακτήρα, το αναγνωριστικό μπορεί να περιέχει και χαρακτήρες '_', οι οποίοι όμως θα πρέπει να μην είναι διαδοχικοί, ενώ αν περιέχει έστω κι έναν χαρακτήρα '_', θα πρέπει και να τελειώνει σε '_'. Για την αναγνώρισή τους χρησιμοποιείται η κανονική έκφραση:

$[a-zA-Z]^+ [0-9]^* ('_' ([0-9]^*([a-zA-Z]^+ \text{---} ([a-zA-Z]^+ '_' [a-zA-Z]^+))))^* '_')^?$

Οι τύποι των σταθερών που υποστηρίζει η γλώσσα είναι οι εξής: ICONST, RCONST, LCONST, CCONST και SCONST. ICONST είναι οι μη προσημασμένες ακέραιες σταθερές, που υποστηρίζουν και δυαδικούς, δεκαεξαδικούς και οκταδικούς αριθμούς. RCONST είναι οι μη προσημασμένες πραγματικές μεταβλητές και ισχύει ότι και με τις ICONST. LCONST είναι οι λογικές σταθερές ('.true.')

ή '.false.'). CCONST είναι οι χαρακτήρες (ASCII 32-126). SCONST είναι οι συμβολοσειρές, αποτελούμενες στην ουσία από χαρακτήρες. Υπάρχουν ακόμα και τα στοιχεία λίστας (LISTFUNC), που είναι ένα "A" και οσαδήποτε "D", ή κανένα "A" και τουλάχιστον ένα "D", μεταξύ "C" και "R".

Συντακτική ανάλυση

Όπως αναφέρθηκε και πιο πάνω, η συντακτική ανάλυση γίνεται επίσης στο αρχείο SimpleFortran.g4. Οι συντακτικοί κανόνες της γλώσσας είναι ίδιοι με αυτούς που ορίζονται στην εκφώνηση της εργασίας. Η μόνη ουσιαστική διαφορά που υπάρχει είναι πως ο κανόνας:

```
expression : expression OROP expression
            | expression ANDOP expression
            | expression RELOP expression
            | expression ADDOP expression
            | expression MULOP expression
            | expression DIVOP expression
            | expression POWEROP expression
            | NOTOP expression
            | ADDOP expression
            | variable
            | simple_constant
            | LENGTH LPAREN expression RPAREN
            | NEW LPAREN expression RPAREN
            | LPAREN expression RPAREN
            | LPAREN expression
            | LPAREN expression COLON expression RPAREN
            | listexpression
```

αντικαταστάθηκε από τον κανόνα:

```
expression : expression OROP expression
            | expression ANDOP expression
            | expression POWEROP expression
            | expression DIVOP expression
            | expression MULOP expression
            | expression ADDOP expression
            | expression RELOP expression
            | NOTOP expression
            | ADDOP expression
            | variable
```

```

| simple_constant
| T_LENGTH L_PAREN expression R_PAREN
| T_NEW L_PAREN expression R_PAREN
| L_PAREN expression R_PAREN
| L_PAREN expression COLON expression R_PAREN
| listexpression

```

Αυτή η αλλαγή έγινε με σκοπό να διορθωθεί η προτεραιότητα των πράξεων. Το εργαλείο Antlr4 θέτει την προτεραιότητα με βάση την πράξη που βρίσκεται πιο πάνω στον κανόνα, όπως εδώ φαίνεται πως την σημαντικότερη προτεραιότητα την έχει η πράξη της δύναμης (POWEROP).

Σημασιολογική ανάλυση και του παραγωγή Πίνακα Συμβόλων

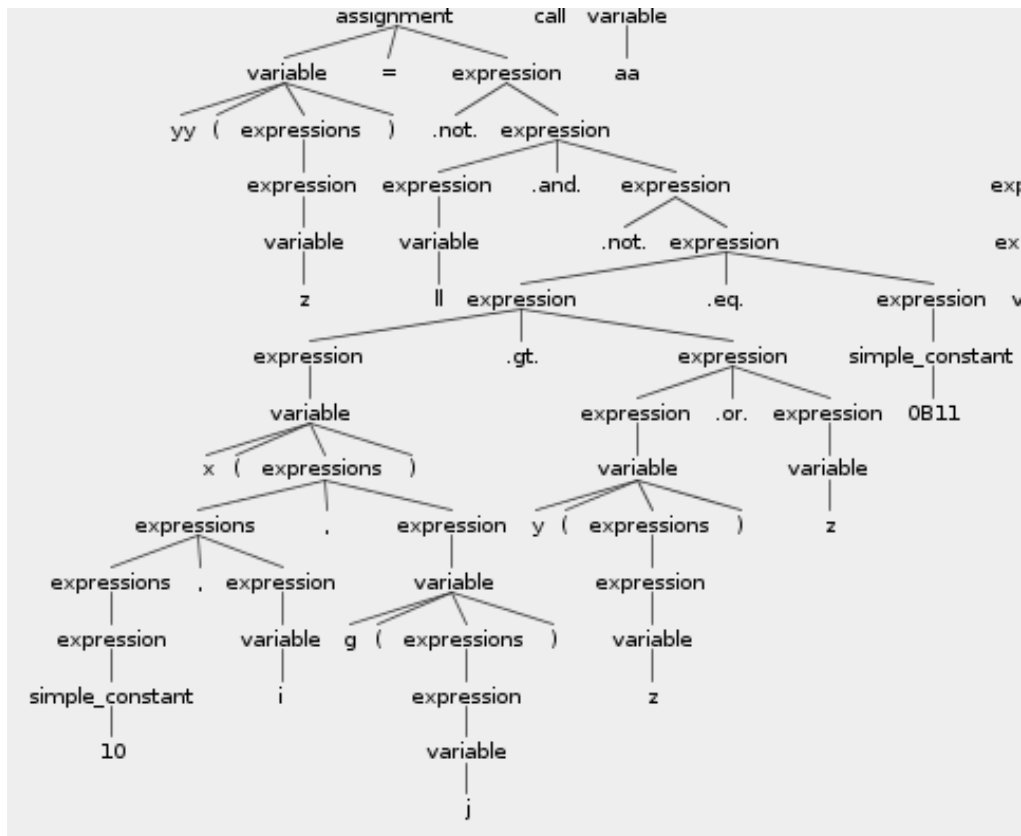
Η σημασιολογική ανάλυση και η παραγωγή του πίνακα συμβόλων γίνεται στα αρχεία Runner.java και Visitor.java. Για να γίνει η σημασιολογική ανάλυση, το εργαλείο Antlr4 παράγει πρώτα το συντακτικό δέντρο για το αρχείο που δέχεται ως input και ορίζεται στην main συνάρτηση της κλάσης Runner. Ένα παράδειγμα μπορούμε να δούμε και αν στον φάκελο που βρίσκονται τα αρχεία τρέχουμε τις εντολές:

```

antlr4 SimpleFortran.g4
javac SimpleFortran*.java
grun SimpleFortran program -gui
</home/dimitris/IdeaProjects/Simple_Fortran/input2.f

```

Το αρχείο "input2.f" που αναφέρεται στην τρίτη εντολή είναι το αρχείο για το οποίο θα παραχθεί το συντακτικό δέντρο. Ένα τμήμα του δέντρου για το συγκεκριμένο αρχείο φαίνεται στην εικόνα:



Για την παραγωγή του πίνακα συμβόλων στην main συνάρτηση της κλάσης "Runner.java" ορίζεται το ParseTreeWalker: walker, που είναι υπεύθυνο για την διάσχυση του δέντρου, η οποία γίνεται με preorder σειρά. Κάθε φορά που το πρόγραμμα επισκέπτεται έναν κόμβο του δέντρου ή φεύγει από κάποιον, τρέχει η κατάλληλη συνάρτηση της κλάσης Visitor.java αν υπάρχει και γίνονται κάποιοι υπολογισμοί, ανάλογα με την περίπτωση. Για παράδειγμα, η συνάρτηση "enterUndef_variable" τρέχει όταν το πρόγραμμα επισκέπτεται τον κόμβο "enterUndef_variable" και γίνεται η αναγνώριση των μεταβλητών που υπάρχουν στα παιδιά αυτού του κόμβου και των τύπων τους. Έπειτα, η main της κλάσης Runner, δέχεται τα αποτελέσματα που έχουν υπολογιστεί στη συνάρτηση Visitor και εμφανίζεται ο πίνακας συμβόλων. Ένα παράδειγμα πίνακα συμβόλων για το αρχείο input1.f είναι:

```

    ### VARIABLES ###
Variables in common regions:
Variable:    i    Region:    com1    Scope:    Global
Variable:    j    Region:    com1    Scope:    Global

Local variables - not in common regions
Integer      x(g)                      Scope:    Local
Integer      z                        Scope:    Local
Integer      i                        Scope:    Local
Integer      j                        Scope:    Local
Integer      z                        Scope:    Local
Integer      n                        Scope:    Local
Integer      a(n)                     Scope:    Local

Logical      l(x)                      Scope:    Local

String       s                        Scope:    Local
String       s_str_(10)               Scope:    Local

Real         lista                    Scope:    Local
Real         i                        Scope:    Local

    ### FUNCTIONS ###

    ### SUBROUTINES ###

Subroutine: try_me_
Parameters:
           integer    a(n)

    ### ERRORS ###

Redeclaration of integer variable: z

```

Παραδείγματα

Τα αρχεία που υπάρχουν στον φάκελο και μπορούν να χρησιμοποιηθούν ως παράδειγμα είναι τα: input1.f, input2.f και test4_ok.f. Για το αρχείο test4_ok.f ο πίνακας συμβόλων είναι:

```

    ### VARIABLES ###
Variables in common regions:

Local variables - not in common regions
Integer      n                Scope:  Local
Integer      x                Scope:  Local
Integer      y(3)             Scope:  Local
Integer      i                Scope:  Local
Integer      number           Scope:  Local
Integer      number           Scope:  Local
Integer      number           Scope:  Local
Integer      number           Scope:  Local

Real         f                Scope:  Local

    ### FUNCTIONS ###

Function: z
Return type: integer
Parameters:
    integer    number

Function: fibonacci
Return type: integer
Parameters:
    integer    number

    ### SUBROUTINES ###

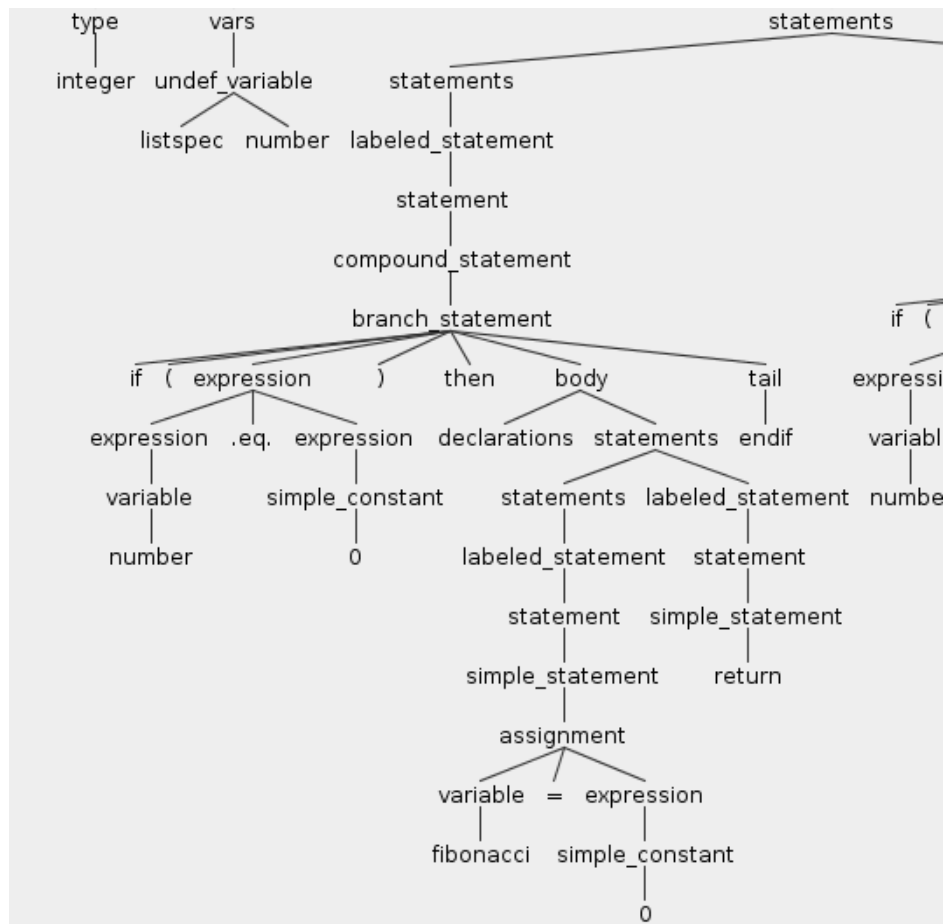
Subroutine: number
Parameters:
    None

    ### ERRORS ###

Redeclaration of integer variable: number

```

Ένα τμήμα του συνανταχτικού δέντρου που παράγεται για το συγκεκριμένο αρχείο είναι:



Για την παραγωγή του πίνακα συμβόλων έτρεξε η main της κλάσης Runner και για την παραγωγή του συντακτικού δέντρου έτρεξαν οι εντολές που αναφέρθηκαν πιο πάνω.