طور تحليل المعاني

في هذا الطور يتم استخدام شجرة الاعراب و معلومات جدول الرموز لفحص توافق المعاني للبرنامج المصدري مع تعريف اللغة و كما يقوم بجمع معلومات النوع و يخزنها في شجرة الاعراب او جدول الرموز . من اهم الوظائف لمحلل المعاني هو عملية فحص النوع Type Checking حيث يقوم المترجم بفحص كل عملية Operator بان معاملاتها متطابقة Matching Operands

في بعض اللغات هناك عملية تدعى Coercion و تعني الاجبار او الاكراه و هي تحويلات قد تسمح بها بعض اللغات (توصيفات اللغة) و ذلك لجعل المعاملات متطابقة , مثلا لو كانت العملية تجري بين نوع صحيح integer و اخر حقيقي floating point فالمترجم يقوم بإجبار coerce العدد الصحيح الى عدد حقيقي .

Type System

مجموعة قوانين لإسناد او تنسيب تعبير النوع Type Expression لمختلف اجزاء البرنامج حيث يتم استخدامها من قبل فاحص النوع Type Checker فاذا كان الفحص يتم اثناء الترجمة فأنه يدعى Static Type Checking اما اذا كان الفحص يتم اثناء التنفيذ فأنه يدعى Dynamic Type Checking

يتألف Type Expression من جزئين :

1- الواصف Constructor (و الذي يطبق على)
2- Basic Type

Example 1\

Array[256] of char

Type Expression array(1..256, char) consists of :

1- Constructor array (that applied to)
2- Subrange [1..256] and type char

Example 2\ ↑ **integer**

Type Expression pointer(integer) consists of :

1- Constructor pointer (that applied to)
2- Type Integer

Example \ suppose you have the following grammar:

P → D ; E

D → D ; D | id : T

T → char | Integer |array[num] of T | ↑T

E → literal | num | id | E mod E | E [E] | E↑

ان الفعالية المرتبطة بالإنتاج ***D → id : T*** تقوم بخزن النوع T في جدول الرموز في المدخل الخاص بالمعرف identifier

addType(id.entry , T.type) applied to entry of id in symbol table and type expression (type) of T.

P → D ; E

D → D ; D

D → id : T { addType(id.entry , T.type)}

T → char {T.type = char}

T→ Integer {T.type = integer}

T → ↑T1 { T.type = pointer(T1.type) }

T → array[num] of T1 { T.type = array(1..num.val , T1.type)}

Type Checking for Expression

E → literal { E.type = char }

E → num { E.type = integer }

E → id { E.type = lookup(id.entry) }

Lookup(e) هي عبارة عن دالة تستخدم لجلب النوع المخزون في جدول الرموز عند المدخل المشار اليه ب e فعندما يظهر معرف في التعبير فأن نوعه المعلن يجلب و ينسب للخاصية Type

E→E1 mod E2 { E.type = If E1.type = integer and E2.type = Integer Then Integer

Else Type_Error }

E→E1[E2] { E.type = If E1.type = integer and E2.type = array(s,t) Then t

Else Type_Error }

E→E1↑ { E.type = If E1.type = pointer(t) Then t Else Type_Error }

Type Checking for Statements

Statements do not have values so we assign basic type (void) to them

S → id := E { S.type = If id.type = E.type Then void Else type_Error}

S → if E then S1 { S.type = If E.type = Boolean Then S1.type

Else type_Error}

S → while E do S1 { S.type = If E.type = Boolean Then S1.type

Else type_Error}

S → S1 ; S2 { S.type = If S1.type = void and S2.type=void

Then void

Else type_Error}