

ANALISIS SEMANTIK

Teknik Kompilasi

Dosen: Utami Dewi W.,S.Kom

Pengertian (1)

Suatu *compiler* selain harus memeriksa leksikal dan sintaks juga harus memeriksa semantik dari program sumber tersebut.

Dalam pemeriksaan semantik yang diperiksa adalah **makna dari setiap pemakaian token** dari program sumber.

Pengertian (2)

Pada tahapan ini dilakukan pengecekan pada struktur akhir yang telah diperoleh dan diperiksa kesesuaiannya dengan komponen program yang ada.

Memfaatkan pohon sintaks yang dihasilkan pada proses parsing. Analisis Semantiks berarti menentukan makna dari serangkaian instruksi yang terdapat dalam program sumber

Lingkup Pemeriksaan

Untuk mengetahui makna maka dilakukan pemeriksaan:

- Apakah variabel yang ada telah didefinisikan sebelumnya, pemeriksaan duplikasi dan terdefinisi.
- Apakah variabel-variabel tersebut tipenya sama, jika ada operasi dua operan maka tipe operan1 harus bisa dioperasikan dengan operan2.
- Apakah operan yang akan dioperasikan tersebut ada nilainya dan seterusnya.

Tipe dan Contoh Kesalahan

- ❑ Contoh Salah menurut semantik (dalam bahasa Pascal)

Program Coba;

 Var a:integer;

 b: string;

begin

 a:=b;

end.

- ❑ Umumnya dalam analisa semantik yang diperiksa adalah kesalahan pemakaian suatu tipe, selain itu beberapa kesalahan lain (static checking), diantaranya **Flow of Control Check, Uniqueness Check, Name Related Check, Type Checking, Type Conversion, Type Coercion.**

Flow of Control Check

Suatu *statement* yang menyebabkan kendali aliran meninggalkan konstruksinya harus mempunyai lokasi untuk memindahkan aliran kendali (*Flow of Control*).

Contoh :

```
{  
    Case 1 : printf ("Satu");  
    break;  
    Case 2 : printf ("dua");  
    Break;  
}
```

Uniqueness Check (1)

Pada situasi tertentu suatu obyek hanya boleh didefinisikan satu kali.

Contoh :

a. Variabel Global tidak boleh didefinisikan > 1 kali

Contoh :

```
Var a: integer; a : string;
```

b. Nama procedure dalam suatu level tidak boleh sama

Contoh :

Uniqueness Check (2)

Procedure Tes;

Begin

end;

Function Tes;

Begin

end;

c. Nama konstanta tidak boleh sama

d. Nama tipe dalam level yang sama tidak boleh sama

Name Related Check (1)

Nama–nama yang sama kadang harus muncul 2 kali. Kompilator memeriksa bahwa nama yang sama tersebut harus muncul 2 kali.

Contoh :

Dalam bahasa dbase, Basic, ADA nama yang sama harus muncul 2 kali sebagai penunjuk akhir.

Dalam dbase

```
For A 1 to 10
```

```
---
```

```
Next A
```

Dalam ADA

```
Procedure Check is
```

```
---
```

```
end Check
```

Name Related Check (2)

Pada beberapa bahasa mereka memiliki *feature* khusus sehingga analisa semantik harus lebih mendalam. Misalnya :

1. Overloaded Operator

Contoh penggunaan tan '+' pada banyak bahasa dapat digunakan untuk tipe bilangan real, integer dan string.

2. Polymorphism

Pada beberapa bahasa diperbolehkan suatu *procedure* punya nama (dalam satu level), asalkan memiliki Jenis berbeda (proc/func) dan Jumlah parameter dan tipenya berbeda (contohnya pada bahasa yang berbasis obyek seperti C++)

Name Related Check (3)

Istilah “polymorphic” juga dapat diterapkan untuk sepenggal code yang dapat dieksekusi dengan argument yang berbeda tipenya.

Contoh :

C reference manual menyatakan tentang pointer operator **&** :

“Jika tipe operand adalah x, maka hasilnya akan bertipe **pointer to x**”.

Karena beberapa tipe data bisa menggantikan x, maka operator **&** dalam C bersifat polymorphic.

Type Checking

Misalnya dalam memeriksa apakah suatu operator digunakan oleh *operand* yang tidak sesuai.

Contoh :

```
Var a: string;  
    b: string;  
begin  
a:= a*b;  
end.
```

Type Conversion (1)

Pada beberapa bahasa dimungkinkan suatu tipe berubah menjadi tipe lain, perubahan ada yang dilakukan secara :

1. Eksplisit

Secara Manual

```
x := ord('A') → byte  
y := chr(32) → char  
m := round(n) → Int
```

2. Implisit

Secara otomatis

```
Var a : real; b : integer;  
Begin  
a:=b/2  
(a := int/2)
```

Type Conversion (2)

Contoh :

Suatu ekspresi adalah $x + i$, dimana : x bertipe real dan i bertipe integer.

Postfix notation untuk $x + i$ adalah :

$x\ i\ \underline{\text{inttoreal}}\ \underline{\text{real+}}$

dimana operator inttoreal meng-konversikan i dari integer ke real, kemudian real+ melakukan operasi penjumlahan.

Type Coercion

Pada bahasa lain suatu tipe dapat dipaksa untuk berubah tipe.

Contoh : dalam bahasa C :

```
Char a;
```

```
Int b;
```

```
b=(int)a;
```

Thank you

