コンパイラ

湯淺太一

第５章

意　味　解　析

Cコンパイラの主要な意味解析処理

1. 名前とオブジェクトとの対応づけ
2. 型チェック

オブジェクト:名前をつけられる構成要素
　変数，関数，ラベル，構造体，共用体，
　構造体/共用体のメンバ，enum定数，型

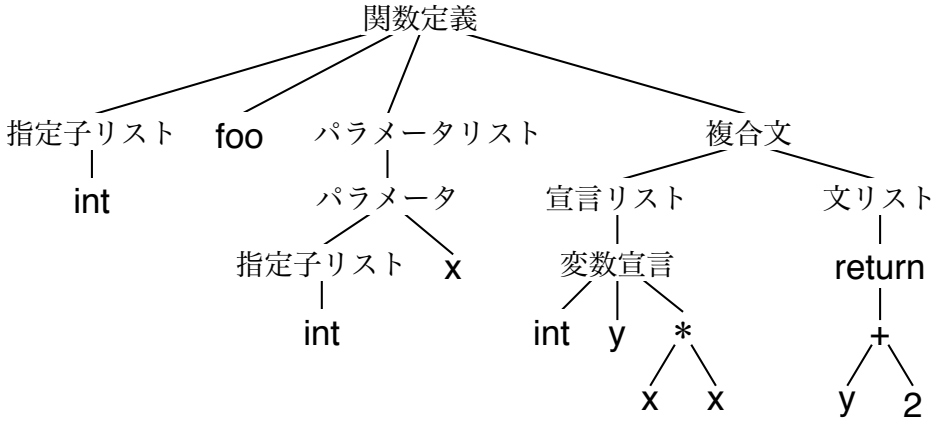
例：

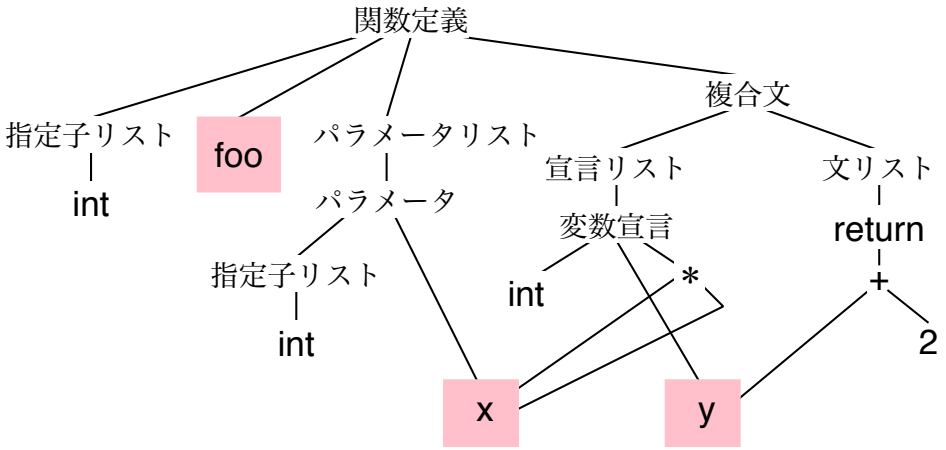
```
int foo(int x) {
    int y = x*x;
    return y+2;
}

double bar(int x) {
    double y = x*x;
    return y+2;
}
```

オブジェクト構造体

オブジェクトを表現する構造体

- オブジェクトの種類(変数，関数，ラベルなど)
- オブジェクトの名前
- 型情報
- 属性(auto，register，staticなど)
- 最適化のための情報
- オブジェクトを割り当てる場所





名前空間とスコープ

名前空間：名前が表すオブジェクトの検索空間

C 言語の主要な名前空間：

- 変数，関数，typedef 名，enum 定数の名前空間
- ラベルの名前空間
- 構造体と共用体の名前空間

例：

```
struct tarao {
    int tarao;
} tarao;

struct tarao x = tarao;
```

スコープ：オブジェクトを参照できる範囲

```
int foo(int x) {
    int y = x*x;
    return y+2;
}
```

```
int foo(int x) {
    int y = x*x;
    return y+2;
}
```

スコープ間の関係

・二つのスコープには共通部分がない.
・二つのスコープはまったく一致する.
・片方のスコープが，もう一方に含まれる.

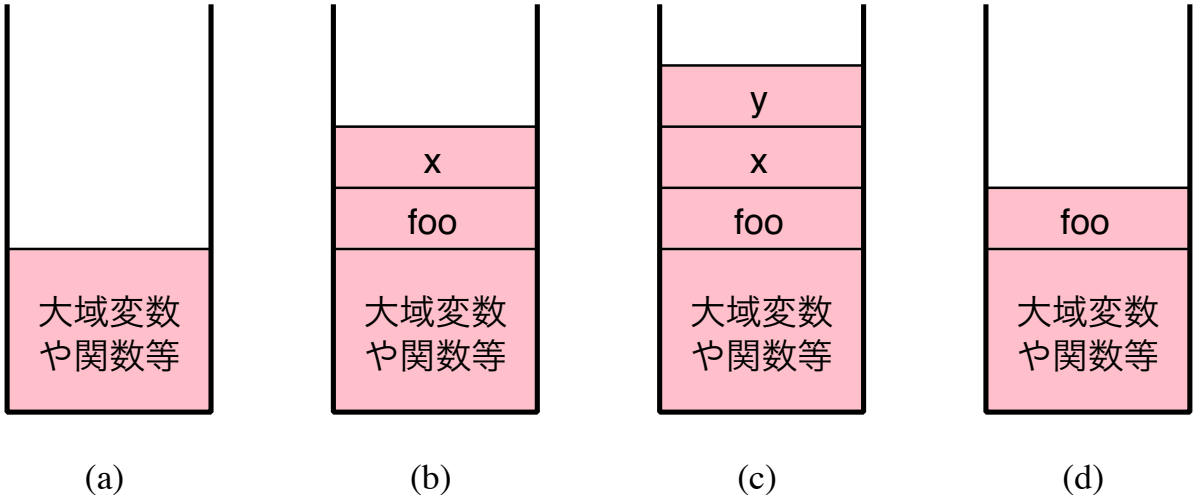
同じ名前空間に属する二つのオブジェクトのスコープが一致するときは，異なる名前をつけなければならない.

例：C 言語の関数パラメータ

名前とオブジェクトの対応づけ

1. まず，xの現れているプログラム中の位置によって，xが属する名前空間を特定する．
2. その名前空間に登録されているオブジェクトのうち，xの現れている位置をスコープに含むものを，内側のスコープを持つものから順に調べる．
3. xを名前とするオブジェクトが見つかれば，最初に見つかったオブジェクトを，xは表す．そのようなオブジェクトがなければ，xの表すオブジェクトは未定義である．

スタックを使った実装



大域変数，関数，typedef名などは，ハッシュ表等に登録するほうが検索効率が良い．

前方参照

```
int sigma(int n) {
    int i = 0, s = 0;
    goto test;
loop:
    s += i;
test:
    if (++i <= n) goto loop;
    return s;
}
```

1. “goto test;” で，未定義ラベル test をラベル表に登録.
   {⟨test，未定義，参照済⟩}
2. “loop:” で，ラベル loop を登録.
   {⟨test，未定義，参照済⟩, ⟨loop，定義済，未参照⟩}
3. “test:” で，ラベル test が「定義済」に.
   {⟨test，定義済，参照済⟩, ⟨loop，定義済，未参照⟩}
4. “goto loop;” で，ラベル loop が「参照済」に.
   {⟨test，定義済，参照済⟩, ⟨loop，定義済，参照済⟩}
5. 関数の意味解析終了時に，ラベル表を調べ，
   「未定義」のラベルがあればエラー，「未参照」のラベルがあれば警告.

型チェックと型変換

型表現：型を表すデータ

⟨char, (signed|unsigned|-)⟩
⟨int, (signed|unsigned|-), (long|short|-)⟩
⟨float⟩
⟨double, (long|-)⟩
⟨enum, ...⟩

多くのプログラミング言語（C 言語を含む）では，構文木を葉から根の方向に向かって調べれば，式の型が決定できる．

例：e_1 * e_2の型チェック
A：e_1の型（型表現）
B：e_2の型（型表現）

```
if (A==Tdouble) {
  check_double(B); return A;
} else if (B==Tdouble) {
  check_double(A); return B;
} else if (A==Tfloat) {
  check_float(B); return A;
} else if (B==Tfloat) {
  check_float(A); return B;
} else {
  check_int(A); check_int(B); return
Tint; }
```