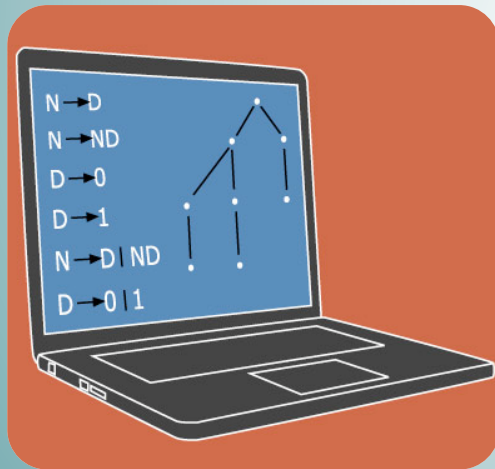


Procesadores de Lenguaje

Ejemplos de código de 3 direcciones



Cristina Tirnauca

Domingo Gómez Pérez

DPTO. DE MATEMÁTICAS,
ESTADÍSTICA Y COMPUTACIÓN

Este tema se publica bajo Licencia:

[Creative Commons BY-NC-SA 3.0](https://creativecommons.org/licenses/by-nc-sa/3.0/)

Código fuente sencillo

Vamos a ver como se traduce de código fuente a código de tres direcciones. El lenguaje fuente tiene las siguientes características:

- ▶ Todos los identificadores son enteros
- ▶ Solo hay las ordenes LEE, ESCR, COPIA EN, MIENTRAS
- ▶ Un identificador puede ser CERO, INCR, DECR

Ejemplo de programa

```
// programa que lee un numero y escribe el doble
lee A
copia A en B
cero C
mientras B
{
  // C es el doble de A-B
  incr C
  incr C
  decr B
}
escribe C
```

Primera Tarea

Escribir un programa en lenguaje fuente para calcular la resta de dos números.

Código de tres direcciones

El código de tres direcciones también es un código simplificado,

- ▶ los registros se denotan $\$p$ y un número,
- ▶ $\$p0$ y $\$p1$ contienen 0 y 1,
- ▶ se tienen solo las operaciones aritmeticas tradicionales tradicionales,
- ▶ hay escribir y leer registros,
- ▶ hay etiquetas para representar los bucles,
- ▶ los saltos son condicionales, solo se salta si el registro es menor o igual que cero.

Ejemplo de código de tres direcciones

```
$p2    :read                    # Read A
$p3    :    $p0  +  $p0        # Set to zero B
loop0:                                # Loop label
$p2    :jump  end0            # Loop condition on A
$p3    :    $p3  -  $p1        # Decrement B
$p2    :write                 # Write A
$p2    :    $p2  -  $p1        # Decrement A
$p0    :jump  loop0          # Go test loop condition
end0:                                # End loop
$p3    :write                 # Write B
                                           # End
```

El intérprete

El intérprete es un programa que ejecuta código de tres direcciones, en este caso está compuesto por:

- ▶ líneas de código (numeradas)
- ▶ un diccionario mem (con los registros)
- ▶ un diccionario label (con las etiquetas)

Segunda Tarea

Escribir un programa en código de tres direcciones para calcular el factorial.

Traducción de un lenguaje a otro

```
statements : statements statement
           |
           ;
statement  : read_stat
           | write_stat
           | incr_stat
           | decr_stat
           | decl_stat
           | copy_stat
           | comp_stat
           | loop_stat
           ;
```

Traducción de un lenguaje a otro

Ahora, en cada regla gramatical añadimos las acciones necesarias,

```
copy_stat : COPIA IDENT EN IDENT
{
    int hs = find_or_error((char *)$2);
    if (hs < 0) yy::parser::error(1,"error");
    int ht = find_or_add((char *)$4);
    sprintf(temp1,"$p%0d",ht);
    sprintf(temp2,"$p%0d",hs);
    sprintf(temp3,"#_Copy_ %_s_ into_ %_s" ,
            (char *)$2, (char *)$4);
    emit(temp1,":",temp2,"+", "$p0",temp3);
};
```

Tercera Tarea

Se dan las siguientes reglas:

`decl_stat` : CERO IDENT

`decr_stat` : DECR IDENT

`incr_stat` : INCR IDENT

¿Que código se debería emitir?

LLVM

LLVM (Low-level virtual machine) es un generador de backends. Dado código en tres direcciones genera código máquina o lo ejecuta directamente. Esta presente en varios compiladores como g++ y clang.

Todo se divide en modulos, dentro de estos modulos se puede:

- ▶ Declarar una función (getOrInserFunction)
- ▶ Declarar bloques (BasicBlock)

Funciones

Las funciones tienen

- ▶ argumentos y tipos de retorno,
- ▶ bloques básicos dentro de ellas.

Bloques básicos

Un bloque básico es un conjunto de instrucciones que solamente hay un salto al final de ellas y se ejecutan **una detrás de otra**. Además, solo puede haber operaciones básicas.

Contexto

El contexto son las variables de entorno que se pueden llamar, aquí se incluyen:

- ▶ Variables globales,
- ▶ variables locales,
- ▶ atributos.