1. Generación de Código Intermedio.

El generador de código intermedio se encarga de convertir la salida producida por el análisis semántico en una representación similar a un lenguaje intermedio que está más cerca del código objeto. Esta representación intermedia requiere dos propiedades importantes. Debe poder crearlo y convertirlo fácilmente en su programa de destino. Existen distintas formas de representación de códigos intermedios, existe la utilización árboles de sintaxis, la notación de postfijo y los códigos de tres direcciones.

Código de tres direcciones

El generador de código intermedio recibe la entrada en forma de un árbol de sintaxis anotado del paso anterior, el analizador semántico. Este árbol de sintaxis se puede convertir en una representación lineal como una notación de postfijo. Por lo tanto, el generador asume que no hay límite en la cantidad de memoria, es decir, registros, para generar el código.

Ejemplo:

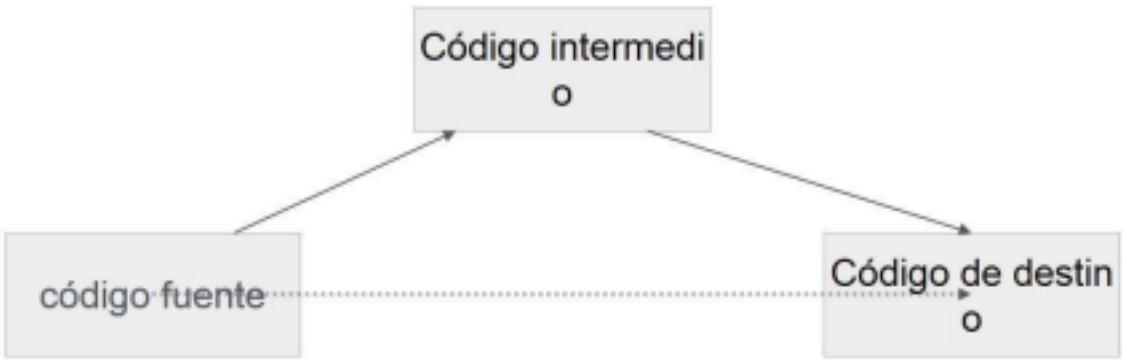
a = b c * d;

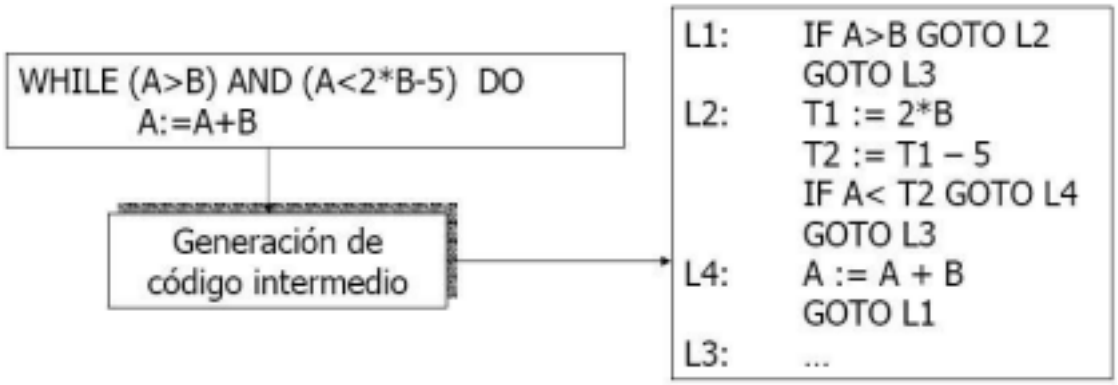
El generador de código intermedio intenta

Y1 = c * d ;. Y2 = b Y1; a = Y2,

Y se utiliza como registro para el programa de destino.

El código de dirección tiene hasta tres posiciones de dirección para calcular la fórmula. Los códigos con tres direcciones se pueden representar de dos formas: cuádruple y triplets.





2. Lenguajes intermedios.

Es un lenguaje utilizado por los compiladores que no es significativo para el SO ni el lenguaje utilizado por el programador para escribir el código a primera instancia, sino que es un lenguaje que actúa como puente entre la creación y la ejecución del programa. Es decir en ocasiones se utiliza un lenguaje intermedio para que un compilador realice optimizaciones muy concretas para que el programa se ejecute de manera más eficiente, pero además se podría utilizar para generar archivos de salida portátiles entre sistema distintos sistemas con incompatibilidad. Obtiene un bytecode por lo que se necesita únicamente el intérprete de esto, Java Virtual Machine.

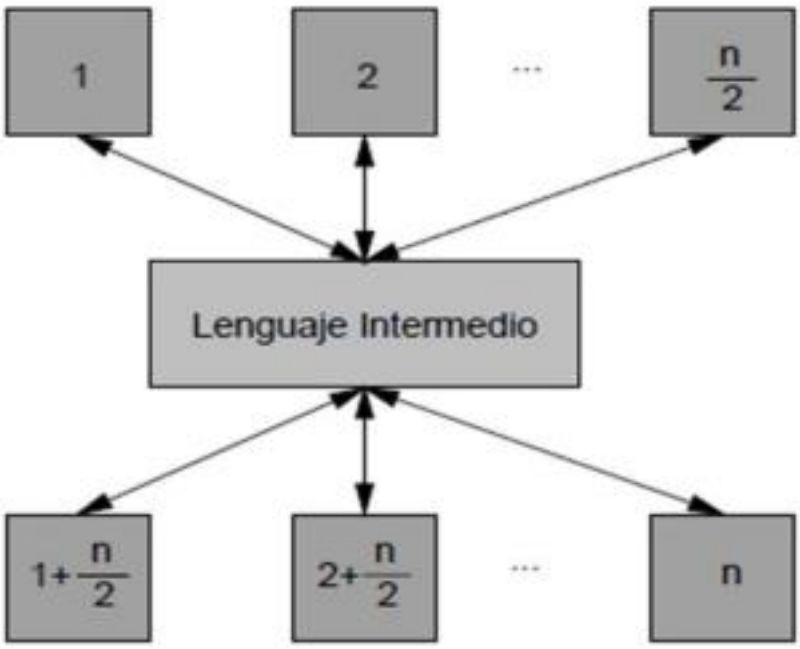


Fig. 32: 2*n traductores entre n lenguajes

Sentencias y expresiones.

· **Expresión:** es una unidad de código que se resuelve en un valor.

Ejemplo: 1+2

· **Sentencias:** se refiere a una instrucción que determina una acción.

```
for (var i=0; i<5; i++) {
// Ejecuta el bloque de código encerrado
}
```

● **De los operandos:** Determinar su localización y efectuar conversiones implícitas.

● **Instrucciones generadas:** Son de acceso a datos, variables simples, registros y vectores.

● **De los operadores:** Estas respetan su procedencia, su asociatividad, orden de evaluación y determinan la interpretación.

● **Manipulación de datos:** De conversiones y operaciones. ● **Flujo de control:** Estas validan el sub-rango y operadores complejos.