

LENGUAJE OBJETO Y MÁQUINA VIRTUAL

Los procesadores de lenguaje producen *programas* en un lenguaje objeto que representa el *significado* de los programas procesados. En su forma más simple, una frase del lenguaje objeto será una secuencia de símbolos o instrucciones.

Los programas en lenguaje objeto adquieren su significado al ser *interpretados* (*ejecutados*) en *máquinas*, que pueden ser virtuales. Una *máquina virtual* es un sistema de transición en un conjunto de estados de cómputo. En nuestro caso se trata de una *Máquina-P* (máquina de pila).

1. LENGUAJE OBJETO: REPERTORIO DE INSTRUCCIONES

A continuación enumeramos las distintas instrucciones del Lenguaje Objeto que vamos a utilizar, especificando sus argumentos y describiendo de manera informal su cometido.

apila(n):

Apila una variable recibida como parámetro.

apila_dir(d):

Apila una variable contenida en la dirección **pos** de memoria.

desapila_dir(d):

Desapila una variable contenida en la pila, guardándola en la dirección **pos** de memoria.

resta

Operación aritmética de resta

suma

Operación aritmética de suma.

or

Operación lógica de or.

multiplica

Operación aritmética de multiplicación.

divide

Operación aritmética de división.

and

Operación lógica de and.

not

Operación lógica de not.

positivo

Cambia de signo un número a positivo.

negativo

Cambia de signo un número a negativo.

mayor

Comprueba si un número es mayor que otro.

menor

Comprueba si un número es menor que otro.

mayor_igual

Comprueba si un número es mayor o igual que otro.

menor_igual

Comprueba si un número es menor o igual que otro.

igual

Comprueba si un número es igual a otro.

distinto

Comprueba si un número es distinto a otro.

stop

Finaliza la ejecución del programa.

En las operaciones aritméticas y lógicas anteriormente citadas, la secuencia de acciones que se lleva a cabo es la siguiente:

- 1- Se desapilan los dos operandos almacenados en la parte superior de la pila.
- 2- Se lleva a cabo la operación sobre esos operandos.
- 3- El resultado obtenido se apila en la cima de la pila.

En el caso de las operaciones de cambio de signo o negación de valor lógico, únicamente se desapila el elemento situado en la cima de la pila, se aplica la operación a dicho elemento y se vuelve a apilar.

2. MÁQUINA VIRTUAL: DESCRIPCIÓN DE LA MÁQUINA-P

Una máquina virtual consiste en:

- Una descripción de los *estados de cómputo*.
- Una descripción de los *estados iniciales*.
- Una descripción de los *estados finales*.
- Un conjunto de *reglas de transición* que relacionan estados con sus *estados siguientes*. Dichas reglas se denotarán como $s \rightarrow s'$.

Componentes:

- Pila de operandos: donde se realizan los cálculos.
- Memoria de Datos: para valores numéricos.
- Memoria de Instrucciones: contiene las instrucciones del programa objeto.
- Contador de Programa (PC).
- Indicador de estado de la máquina.

Estados de la máquina virtual:

Son tuplas de la forma $\langle P, S, M, CP, E \rangle$ donde:

P: Programa.

S: Pila (stack).

M: Memoria de Datos (RAM).

CP: Contador de programa.

E: Estado de la máquina. Puede valer r (máquina en ejecución), s (máquina parada), o e (error).

El estado inicial consiste en un programa a ejecutar; la pila y la memoria de datos vacías, el contador de programa a cero y en estado de ejecución: $\langle P, [], \{ \}, 0, r \rangle$

Los posibles estados finales se dan cuando la máquina se para o hay un error:

$\langle P, S, M, CP, s \rangle$ y $\langle P, S, M, CP, e \rangle$

Reglas de transición de estados y definición formal de las instrucciones del lenguaje objeto.

A continuación muestran todas las posibles transiciones de estados de nuestra máquina virtual, usando la siguiente notación:

- $[]$ pila vacía.
- $[v | R]$ pila con cima v y resto R.
- $[v_0, v_1 | R]$ pila con cima v_0 , subcima v_1 y resto R.
- $I(P, i)$ i-ésima instrucción de un programa P.
- $M(d)$ contenido de la posición d de la Memoria de Datos.
- $M(d \leftarrow v)$ asignar el valor v a la posición d de la Memoria de Datos.
- OpArit operadores aritmético-lógicos: + (suma), - (resta), * (multiplica), / (divide), and, or.
- OpRel operadores relacionales: > (mayor), < (menor), >= (mayor_igual), <= (menor_igual), = (igual), != (distinto)

Transiciones:

| | | | |
|--|---------------|---|---|
| $\langle P, [v_0, v_1 R], M, i, r \rangle$ | \rightarrow | $\langle P, [v_1 \text{ OpArit } v_0 R], M, i+1, r \rangle$ | si $I(P, i) = \text{OpArit}$ |
| $\langle P, [v_0 R], M, i, r \rangle$ | \rightarrow | $\langle P, [\text{not } v_0 R], M, i+1, r \rangle$ | si $I(P, i) = \text{not}$. |
| $\langle P, S, M, i, r \rangle$ | \rightarrow | $\langle P, [n S], M, i+1, r \rangle$ | si $I(P, i) = \text{apila}(n)$. |
| $\langle P, S, M, i, r \rangle$ | \rightarrow | $\langle P, [M(d) S], M, i+1, r \rangle$ | si $I(P, i) = \text{apila_dir}(d)$. |
| $\langle P, [v_0 R], M, i, r \rangle$ | \rightarrow | $\langle P, [R], M(d < -v_0), i+1, r \rangle$ | si $I(P, i) = \text{desapila_dir}(d)$. |
| $\langle P, [v_0 R], M, i, r \rangle$ | \rightarrow | $\langle P, [v_0 R], M, i, r \rangle$ | si $(I(P, i) = \text{negativo y } v_0 \leq 0)$ o $(I(P, i) = \text{positivo y } v_0 \geq 0)$ |
| $\langle P, [v_0 R], M, i, r \rangle$ | \rightarrow | $\langle P, [-v_0 R], M, i, r \rangle$ | si $(I(P, i) = \text{negativo y } v_0 > 0)$ o $(I(P, i) = \text{positivo y } v_0 < 0)$ |
| $\langle P, [v_0, v_1 R], M, i, r \rangle$ | \rightarrow | $\langle P, [0 R], M, i+1, r \rangle$ | si $I(P, i) = \text{OpRel}$ y $(v_0 \text{ OpRel } v_1) = \text{falso}$ |
| $\langle P, [v_0, v_1 R], M, i, r \rangle$ | \rightarrow | $\langle P, [1 R], M, i+1, r \rangle$ | si $I(P, i) = \text{OpRel}$ y $(v_0 \text{ OpRel } v_1) = \text{cierto}$ |
| $\langle P, S, M, i, r \rangle$ | \rightarrow | $\langle P, S, M, i, s \rangle$ | si $I(P, i) = \text{stop}$. |
| $\langle P, S, M, i, r \rangle$ | \rightarrow | $\langle P, S, M, i, s \rangle$ | si $I(P, i)$ está indefinida. |
| $\langle P, S, M, i, r \rangle$ | \rightarrow | $\langle P, S, M, i, e \rangle$ | en cualquier otro caso. |