S-> sumasi(O id, id; L)

L-> L, cte

L-> cte

O -> < | > | <= | >= | <> | ==

sumasi ( < b, var; 10, 20 , 30 )

R02 O-><>

R04 L->cte

R03 L->L, cte

R03 L->L, cte

R01 S-> sumasi(O id, id; L)

Polaca

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| @a | @b | CMP | BNE | 24 | 10 | @b | = | 20 | @b | = | BI | 01 | 30 | @b | = | BI | 01 | | b | @b | = | BI | 01 | var | var | @aux | @b | + | = |  |
| R02 O->> | | | | | R04 L->cte | | | R03 L->L, cte | | | | | R03 L->L, cte | | | | | R01 S-> sumasi(O id, id; L) | | | | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |
| 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |
|  | | | | |  | | |  | | | | |  | | | | | |  | | | | | | | | | | | |

|  |  |
| --- | --- |
| R01 S-> sumasi(O id, id; L) | i\_pol(b), @b=b@, i\_pol(BI),  x=desapilar(tope-6), z=x,  h=desapilar(tope), h=x,  i=desapilar(tope-2), i=x  x=desapilar(tope-3), x=celda actual+1,  i\_pol(var), i\_pol(var), i\_pol(@b), i\_pol(+),i\_pol(=), |
| R02 O->> | apilar()i\_pol(@a), i\_pol(@b), i\_pol(CMP), i\_pol(BLE), apilar(), avanzar() |
| R03 L->L, cte | apilar(), i\_pol(cte), i\_pol(BI), apilar(), avanzar() |
| R04 L->cte | i\_pol(cte), |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| @a  @b=10 | @b=10 | @b=20 | @b=30 | @b=b  Celda 17=@a  Celda 5=18 |  |
| O->> | L->cte | L->L, cte | L->L, cte | S-> sumasi(O id, id; L) |  |

PILA

|  |  |  |
| --- | --- | --- |
| O->>, L->cte, L->L, cte, L->L, cte | S-> sumasi(O id, id; L) | |
|  |  |  |
| BI del 30 |  |  |
| @principio30 |  |  |
| BI del 20 |  |  |
| @principio20 | @principio30 | @principio30 |
| @bi del O-> | @principio20 | @principio20 |
| @a | @bi del O-> | @bi del O-> |
| $ | $ | $ |

Asi y todo todavía falta analizar los otros dos elementos

Osea después de entrar a la comparación BNE habría que agregar un else para que apunte el puntero de la pila de principio de 20 el cual ejecuta la asignación de @b y vuelve a preguntar @a, @b

Si se da cam,ina a la 24 y hace la suma y asignación y sino debería ir a el else metiendo el principio de 30 y saltar y que 30 asigne en @b y vuelva a l principio de todo para que se compare @a y @b.

En elc aso de que estos elementos sean negativos también debería a final de todo el código estar los mismos saltos del “else” de la condición con el salto a cada elemento de la lista usando los mismos punteros de la pila sin eliminarlos

La complejidad de este ejercicio es altísima para un simple curso de compiladores