Three address code for one dimentional array

If the width of each array element (or size of the element for example integer has a size of 2 bytes) is w, then the ith element of A has a address or a location

= base address + $i * w_i$

where base address is a constant term & $i * w_i$ is a variable term.

So, Three address code for a[i] is

① t_1 = a[constant]
② t_2 = $i * w_i$
③ t_3 = t_1[t_2]

For example convert the statement x = a[i] + b[j] is

t_1 = a[constant]
t_2 = $i * w_i$
t_3 = t_1[t_2]
t_4 = b[constant]
t_5 = $j * w_j$
t_6 = t_4[t_5]
t_7 = t_3 + t_6
x = t_7

Three address code for two dimentional array

address in row major representation of a[i,j] = base address + ((i - low_1) x n_2 + (j - low_2)) * w

where a[i,j] means address of ith element in the jth row.

low_1 & low_2 are two lower bound on value i and j

n_2 is a number of value that j can take or number of element in the array a[i,j] i.e

n_2 = $high_2$ - low_2 + 1

or $n_2 = u_2 - l_2 + 1$

u_2 is upper bound &

l_2 is lower bound

now,

address of a[i,j] can be written as

$= \text{base address} + (i*n_2 - low_1 n_2 + j - low_2) w$

$= \text{base address} + i*n_2*w - low_1 n_2 w + j*w - low_2 w$

$$= \underbrace{\text{base address} - ((low_1 * n_2) + low_2) * w}_{\text{constant term}} + \underbrace{((i \times n_2) + j) * w}_{\text{Variable term}}$$

So three address code of a[i,j] can be computed as

$t_1 = i * n_2$
$t_2 = t_1 + j$
$t_3 = t_2 * |w|$
$t_4 = a[constant]$

$t_5 = t_4[t_3]$

For example consider the following code of segment

p = 0
i = 1, j = 1
do
p = p + a[i, j]
i = i + 1
j = j + 1
while (i ≤ 10)

1. p = 0
2. i = 1
3. j = 1
4. $t_1 = i * n_2$
5. $t_2 = t_1 + j$
6. $t_3 = t_2 * |w|$
7. $t_4 = a[constant]$
8. $t_5 = t_4[t_3]$
9. $t_6 = p + t_5$
10. $p = t_6$
11. $t_7 = i + 1$
12. $i = t_7$
13. $t_8 = j + 1$
14. $j = t_8$
15. If (i ≤ 10) goto 4

Scanned with CamScanner