Assingment-1
Thanjida Akhter
201691489

Solution: 1

a)

Suppose there are two matrixes A and B Both are 4X4 matrixes. There multiplication is C. I give an example how matrix multiplication work, after then we convert it to binary tree.

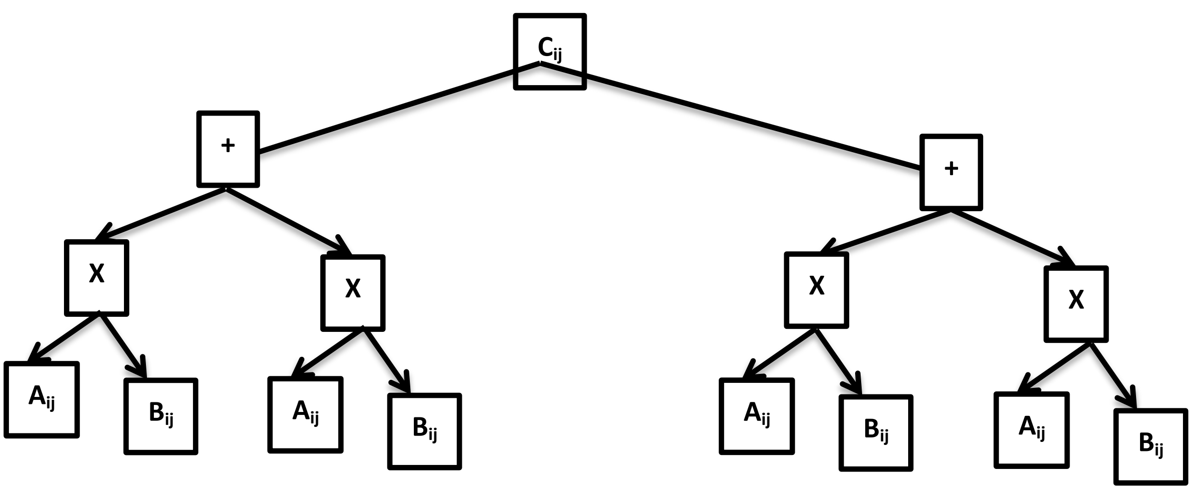
$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}, \qquad B = \begin{pmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \\ b_{41} & b_{42} & b_{43} & b_{44} \end{pmatrix}$$

$$C = AxB = \begin{pmatrix} C_{11} & C_{12} & C_{13} & C_{14} \\ C_{21} & C_{22} & C_{23} & C_{24} \\ C_{31} & C_{32} & C_{33} & C_{34} \\ C_{41} & C_{42} & C_{43} & C_{44} \end{pmatrix}$$

Here only,

C_{11}		$a_{11b_{11}}$	$+a_{12}b_{21}$	$+a_{13}b_{31}$	$+a_{14}b_{41}$
C_{12}	=	$a_{21}b_{11}$	$+a_{22}b_{21}$	$+a_{23}b_{31}$	$+a_{24}b_{41}$
C_{31}		$a_{31}b_{11}$	$+a_{32}b_{21}$	$+a_{33}b_{31}$	$+a_{34}b_{41}$
C_{41}		$a_{41}b_{11}$	$+a_{42}b_{21}$	$+a_{43}b_{31}$	$+a_{44}b_{41}$

We can see all calculation is similar so we create one binary tree which represents nxn matrix multiplication. Here i=row number, j= column number.



b)

Suppose we have two matrixes both are 4x4 they are A= (a_{11} ......a_{44}) and B= (b_{11} ......b_{44}), in our pc we take 16 register like R_{11} .....R_{44}. In super-scalar processors in one time only two instructions will work also we can use lots of register.

1. Load two values from two matrix a_{11} and b_{11}
2. Multiply a_{11}, b_{11} store it in C_1 and load a_{12}
3. Load b_{21} and multiply a_{12}, b_{12} and store it in C_2
4. Load a_{13} and load b_{31}
5. Multiply a_{13}, b_{31} store C_3 and load a_{14}
6. Load b_{41} and multiply a_{14}, b_{41} store in C_4
7. Add C_1 and C_2 store it C_1 and Add C_3 and C_4 and store it C_2
8. Add C_1 and C_2 store it R_{11}

For other calculation follow the same instructions only need to change the row column value.

c)

I my pseudocode there are 8 load, 4 multiplications and 3 add. Same instructions for 16 times.

Execution time= 16*(8*1 +3* 1 +4* 8)=688

Solution: 2

a)

Here we <x=x+1> is an atomic operation so it's not visible to other processes until it is finished. For x=x-y; Here y is being referred to process y=0; which is a critical reference and x is being written by the first process.

So it's not hold at most once property. Because x is also being written by both processes at the same time, x in <x=x+1> is invisible to other processes writing the value of x in process 1 is sharing the common variable x.

b)

Parallel running three threads at a time, so here 3! =6 Combinations.

Suppose at first x=1 and y=1, then the possible value for x and y given bellow:

1	x = x + 1	y = 0	x = x – y	x=2, y=0
2	x = x + 1	x = x – y	y = 0	x=1, y=0
3	y = 0	x = x + 1	x = x – y	x=2, y=0
4	y = 0	x = x – y	x = x + 1	x= 2, y=0
5	x = x – y	y = 0	x = x + 1	x=1, y=0
6	x = x – y	x = x + 1	y = 0;	x=1, y=0

Solution: 3

a)

In repeat rule, X is containing repeat and Y is containing the program after repeat.

$$\frac{(\mathrm{X}, \mathrm{s}) \ \rightarrow s'}{(if\ \mathrm{b}\ then\ \mathrm{X}\ else\ \mathrm{Y}, s\ ) \ \rightarrow S(\mathrm{X}, s')}$$

$$if\ B[[b]] = True$$

$$\frac{(\mathrm{Y}, s) \ \rightarrow s'}{(if\ b\ then\ \mathrm{X}\ else\ \mathrm{Y}, s\ ) \rightarrow S(\mathrm{Y}, s')}$$

$$if\ B[[b]] = False$$

$$\frac{\begin{array}{l}\{P \wedge \ulcorner\ B\}\ S\ \{Q\} \quad Q \rightarrow (P\ \wedge B) \\ (S(B) = False;\ while\ \ulcorner B\ do\ S\ else\ skip, s\ ) \\ \rightarrow (if\ \ulcorner B\ then\ S;\ (while\ \ulcorner B\ do\ S,\ else\ skip, s'))\end{array}}{\{\mathrm{X} \wedge \ulcorner\ B\}\ repeat\ S\ \{\mathrm{X} \wedge \ulcorner\ B\}}$$

$$\frac{\begin{array}{l}S(B) = True\ (if\ B\ then\ \mathrm{X}\ else\ \mathrm{Y}, s) \ \rightarrow S(\mathrm{X}, s)\ S[b \mapsto False] \\ S(B) = False\ (if\ B\ then\ \mathrm{X}\ else\ \mathrm{Y}, s) \ \rightarrow S(\mathrm{Y}, s)\ S[b \mapsto True]\end{array}}{(while\ B\ do\ S\ else\ skip, s) \ \rightarrow (if\ b\ then\ S;\ (while\ b\ do\ S),\ esle\ skip, s')}$$

b)

For while,

$$\frac{\{\mathrm{X}\ \wedge\ B\ \}\ S; \{Q\} \quad B\ \Longrightarrow true \quad P\ \Longrightarrow (\mathrm{X}\ \wedge\ B\ )\ Q\ \Longrightarrow (\mathrm{X}\ \ \wedge \ulcorner B)}{\{\mathrm{Y}\}\ while\ B\ do\ S\ \{Q\}} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots(i)$$

For Repeat,

$$\frac{\{\mathrm{X}\ \wedge\ B\ \}\ S; \{Q\} \quad P\ \Longrightarrow\ \{\ \mathrm{X}\ \wedge\ \ulcorner B\ \} \quad Q\ \Longrightarrow (\mathrm{X}\ \ \wedge\ \ B)}{\{P\}\ Repeat\ S\ for \ulcorner\ B\ \ \{Q\}} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(ii)$$

Defining while(!B) in (iii) from (i) and (ii)

$$\frac{\{\mathrm{X}\ \wedge\ B\ \}\ S; \{Q\} \quad P\ \Longrightarrow\ \{\ \mathrm{X} \wedge\ \ulcorner B\ \} \quad Q\ \Longrightarrow (\mathrm{X}\ \wedge\ \ B)}{\{P\}\ While \ulcorner\ \ B\ do\ S\ \{Q\}}$$

From (ii) and (iii) we came to know that repeat is equivalent to while (!B)