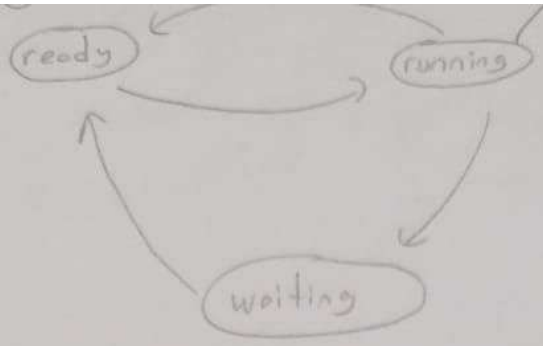
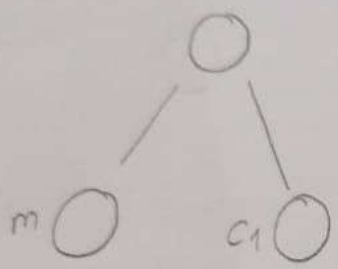
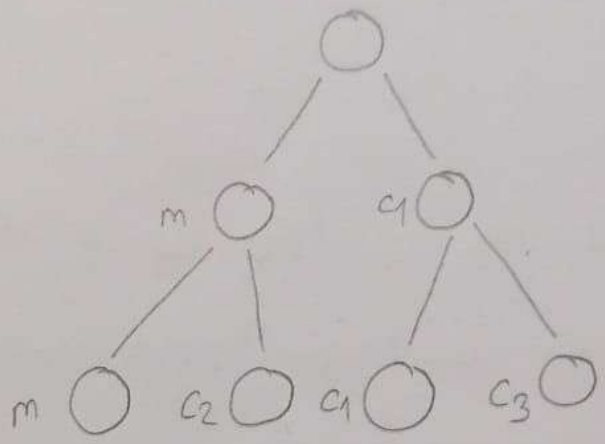


(7)

n = 1

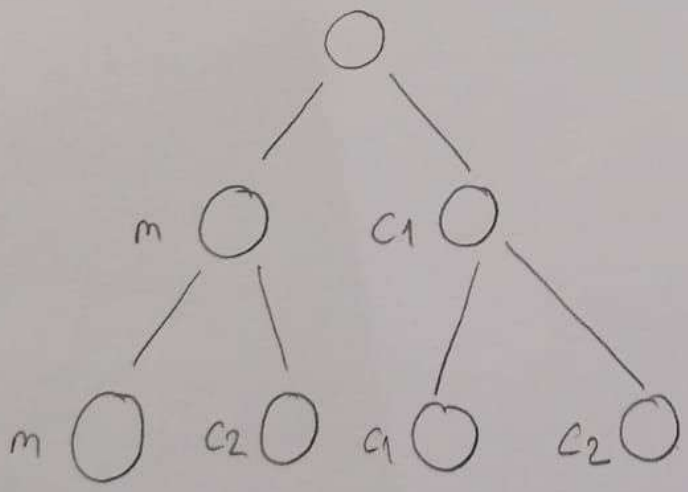


n = 2



$2^n - 1$ child

(8)



C1, C2, C3

(9)

a) $0,1 \times 0,9 = 0,09$

Overall time will be reduced 1-9

Overall speed up = $\frac{1}{0,91}$

b) $\frac{1}{(1-0,60) + \frac{0,60}{30}}$ $\frac{1}{0,42}$

(11) Deadlock

P_1 wait P_0 , P_0 wait P_1

They both wait for each other forever.

(12)

```
monitor    sumUniquenumbers {

    enum { THINKING; WAITING, READING } state[n];
    Condition  self[n];
    int  total;
    void  open(int i) {
        state[i] = WAITING;
        if (i + total => n)  self[i].wait();
        state[i] = READING
        total += i    }

    void  close(int i) {
        state[i] = THINKING;
        total -= i;
        for (int j = n - total - 1; j => 0; j--) {
            if (state[j] == WAITING {
                    self[j].signal(); break; }
        }

    initialization.code() {

        for (int i = 0; i < n; i++) {
            state[i] = THINKING;
    }

}
```

(13)

```
Semaphore fullBuffers = 0
Semaphore emptyBuffers = n
Semaphore mutex = 1

Producer() {

    wait(empty);
    wait(mutex);

put 1 coke in machine

    signal(mutex);
    signal(full);

}

Consumer() {

    wait(full);
    wait(mutex);

take 1 coke out.

    signal(mutex);
    signal(empty);

}
```