```
Pseudo Algorithm:
Extended Reachability Graph Generation from GSPN:

S: Stack of not processed Markings
RG: current reachability graph

S.push(m_init)
RG.insert_state(m_init,0,0)
While not_empty(S)
    m=S.pop
    vanishing = FALSE

    //first fire the immediate transitions
    Forall Transitions t
        if (enabled(t,m) && immediate(t))
            vanishing = TRUE
            process(m,t)
        endif
    endfor

    //if tangible state, fire the timed transitions
    if (NOT vanishing)
        Forall Transitions t
            if (enabled(t,m) && timed(t))
                process(m,t)
            endif
        endfor
    endif
endwhile

//process one marking+firing
process(m,t){
    m_next = fire(t,m)
    m_found = RG.find(m_next)
    if (m_found == NULL)
        RG.insert_state(m_next,m,t)
        S.push(m_next)
    else
        RG.connect(m,m_found,t)
    endif
}

MC.insert_state(m1, m2, r)
- inserts a new state(m1) into the CTMC and make an arc from the predecessor (m2) wi
th the given rate r

MC.connect(m1, m2, r)
- make an arc from the predecessor (m2) to the marking (m1) with the given rate r

MC.find(m)
- if marking m is found in the CMTC, then it returns the corresponding object, retur
ns NULL

enabled (t, m)
- returns true if transition t is enabled in marking m

fire (t,m)
- returns the marking, that resuklts when transition t fires in marking m
```