

Pseudo Algorithm:

CTMC Generation from a GSPN:

S: Stack of not processed Markings

MC: current markov chain

//assuming the initial marking is tangible!

S.push(m_init)

MC.insert_state(m_init,0,0)

While not_empty(S)

 m=S.pop

 Forall Transitions t

 if (enabled(t,m))

 process(t,m,t.rate)

 endif

 endfor

endwhile

//recursive processing of the petri net markings

process (t,m,rate){

 m_next = fire(t,m)

 vanishing = FALSE

 sum = 0

 //find sum of probabilities of all enabled immediate transitions

 Forall Transitions ti

 if (immediate(ti) && enabled(ti,m_next))

 sum += ti.prob

 vanishing = TRUE

 endif

 endfor

 //skip vanishing marking, if any existed

 if (vanishing)

 Forall Transitions ti

 if (immediate(ti) && enabled(ti,m_next))

 process(ti,m_next,rate*ti.prob/sum)

 endif

 endfor

 endif

 //insert tangible marking

 if (NOT vanishing)

 m_found = MC.find(m_next)

 if (m_found == NULL)

 MC.insert_state(m_next,m,rate)

 S.push(m_next)

 else

 MC.connect(m_found,m,rate)

 endif

 endif

}

MC.insert_state(m1, m2, r)

- inserts a new state(m1) into the CTMC and make an arc from the predecessor (m2) with 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 CTMC, then it returns the corresponding object, returns NULL

enabled (t, m)

- returns true if transition t is enabled in marking m

fire (t,m)

- returns the marking, that results when transition t fires in marking m