Problem Set 8

Because we have a directed tree topology, we have for each node i r(i), and for consistency let's assume the root reads from a special channel which is empty this way every node reads from some channel

ASSUMPTION: For simplicity lets define reachability such that we are interested in every process reaching a target state

Given the reachability problem i.e the configuration of states that need to be reached in a subset of process. We define the below languages Let $P \in \mathscr{P}$ be the set of processes whose target state reachability we are interested in. We define the following automaton for them using their transition systems. For every $i \in \mathscr{P}$ we define $A_i = (Q_i, \Sigma_i, \delta_i, s_i, F_i)$

where Q_i are the same as in the transition system, $\Sigma_i =$ union of channel alphabets for which this process is a reader and writer, δ_i is the same as in the transition system but the operation replaced by the message letter being read or written and no-op replaced by ϵ s_i is the same as transition system initial state F_i is the target state we are interested in if $i \in P$ else it is Q_i

L_i is the language accepted by the automaton A_i

$L_i{\downarrow_r}$ is the language where only symbols from the reader channel are present, i.e other alphabets are replaced by epsilon.

ASSUMPTION : w.l.o.g we assume that all the channel alphabets are disjoint ASSUMPTION : additionally lets say we removed all the epsilon transitions and have a DFA

After we do all of this a run it is clear that we can go from a run in the global configuration graph to a run in this system, basically the channels are removed that's all right.

But note that a particular transition involving an alphabet from reader channel for that process is only possible if that alphabet was at the head in the reader channel

Given the directed tree topology, we define a language L_i^e as follows: If process i is a leaf then $L_i^e = L_i \cap M^*_{r(i)}$ where $r(i)$ is the channel that process i reads from.

If process i is a non-leaf, and the children of i are $k_1, k_2, ..., k_m$ then $L_i^e = L_i \cap shuffle(L^e_{k_1}{\downarrow_r}, L^e_{k_2}{\downarrow_r}, ..., L^e_{k_m}{\downarrow_r}, M^*_{r(i)})$

1 Lemma1

Consider the tree topology

Lemma1 : For any word w in L_j^e with channel r(i) supplying the $w \downarrow_{r(i)}$ then we can form a run from (s0,s1,...sn) to (f0,f1,...fn) where 0...n are the nodes in the subtree rooted by j

Proof2: Induction on the no. of nodes in the directed tree topology

Base case: L_j^e is ϵ which is non empty, imples $\epsilon \in L_j$, implies $s_j = f_j$, clearly since we start in s_j anyway it is reachable

Induction: Let i be a non-leaf node and let the children be $k_1, k_2, ..., k_m$, L_r^e is non-empty implies $L_i^e = L_i \cap shuffle(L^e_{k_1}{\downarrow_r}, L^e_{k_2}{\downarrow_r}, ..., L^e_{k_m}{\downarrow_r}, M^*_{r(i)})$ is non-empty

let $w \in L_r^e$ w is of the form $shuffle(w_1, w_2, ..., w_m)$

$w_i \downarrow_{r(k_i)} = w_i' \downarrow r(k_i)$ where $w_i' \in L^e_{k_i}$

By induction hypothesis if channel $r(k_i)$ supplies the word $w_i' \downarrow r(k_i)$ then we can generate run where nodes in the tree rooted by k_i all reach their target states.

Since $w \in L_r^e$ does supply it, we can form this run. Similarly we can form runs for nodes in subtrees rooted at other children

By concatenating the runs we can form the final run that we want.

Theorem 1: Reachable(there exists a run such that the target states are reached in the given topology) iff L_r^e is not empty (Where r is the root process in the topology)

($\leftarrow$) Since root does not need anything to be supplied in its channel, by using lemma 1

($\rightarrow$)

2 Lemma 2

Claim 1: Given a directed tree topology with r=0 as the root of the tree.

$w \in L_r^e \implies (s_0, s_1, ..., s_n, w\downarrow_{r(0)}, \epsilon, ..., \epsilon) \rightarrow_G^* (f_0, f_1, ..., f_n, \epsilon, \epsilon, ...\epsilon)$

Proof1: Induction on the no. of nodes in the directed tree topology

Base case: We have only one node r, $L_r^e = \epsilon \implies \epsilon \in L_r \implies s_r = f_r$, clearly since we start in s_r it is reachable

Induction: Let r be a non-leaf node and let the children be $k_1, k_2, ..., k_m$, $w \in L_r^e \implies w \in L_i \cap shuffle(L_{k_1}^e\downarrow_r, L_{k_2}^e\downarrow_r, ..., L_{k_m}^e\downarrow_r, M_{r(i)}^*)$ w is of the form $shuffle(w_1, w_2, ..., w_m)$ where $w_i \in L_{k_i}^e\downarrow_{r(i)}$

$w_i \downarrow_{r(k_i)} = w_i' \downarrow r(k_i)$ where $w_i' \in L_{k_i}^e$

Let π_i be the tree that is rooted by k_i By induction hypothesis, since $w_i' \in L_{k_i}^e$ we have run from nodes in π are in start state, and reader channel of k_i has $w_i' \downarrow r(k_i)$ to a final configuration where every node is in the final state and all channels are empty.

Since w is in L_r^e it means it is also in L_r we can take this run and convert to the corresponding run in the configuration graph where only r is moving, and this w can generate $w_i' \downarrow r(k_i)$ in each k_i's reader channel, so we can start from $(s_0, s_1, ..., s_n, \epsilon ...)- >$ (only r moves and fills up all the channels)$- > (f_0, s_1, ..., s_n, \epsilon, w_i \downarrow_{r(k_i)}, ...w_i \downarrow_{r(k_i)})- >$ $stitchthek_irunoneaftertheother$

Claim 2: Given a directed tree topology with i=0 as the root of the tree.

$(s_0, s_1, ..., s_n, \alpha, \epsilon, ..., \epsilon) \rightarrow_G^* (f_0, f_1, ..., f_n, \epsilon, \epsilon, ...\epsilon) => \exists w \in L_i^e, \alpha\downarrow_{r(0)} = w\downarrow_{r(0)}$

Proof2:

Base case : There is only one node α has to be epsilon

Induction: Let r be a non-leaf node, and $k_1, k_2, ..., k_n$ be the children of the r and let the tree rooted by them be $\pi_1, \pi_2, .., \pi_n$

Let r run and fill all the reader channels with α_i For each k_i we form a run from where only nodes in π_i are in start state and α_i is in the reader channel of k_i, there and final state is reached. Now we use induction hypothesis to say that there is a word in $w_i \in L_{k_i}^e$ and $\alpha_i \downarrow_{r(0)} = w_i \downarrow_{r(0)}$

So we find a w_i for each k_i

so we have $w_i \downarrow_{r(i)} \in L_i^e$ so we have $\alpha_i \in L_i^e$

Consider the word $w \in shuffle(\alpha_1, \alpha_2, ...\alpha_n, \alpha)$ $walsobelongstoL_r$

$w \in L_r^e$ and $w \downarrow_{r(r)} = \alpha$