A Report on

A Banker's Solution for Deadlock Avoidance in FMS With Flexible Routing and Multiresource States

Shivam Bhagat 2015B5A70460H
Abhinav Kumar 2015B5A70674H
Yashdeep Thorat 2015B5A70675H

The paper is based on a popular formal method, Petri net which is commonly used to model Flexible Manufacturing Systems (FMS). Petri nets are used generally to model the dynamic behaviour of any system. These nets are basically a directed bipartite graph with the set of vertices/nodes as "Places" & "Transitions" and the edges are called "Arcs". Places can be thought of as storehouses of entities known as "Tokens". It is these tokens that bring in the dynamic nature to this static net. The set of tokens present in each place at a particular time is called a marking. The markings keep on changing with time. A reachable marking is one which can be obtained by a series of transitions. The authors introduce a novel class of Petri nets, named S^*PR which is a generalisation of previously introduced classes like S^3PR and S^4PR. This paper addresses the problem of deadlock and one of its solution in FMS modelled by S^*PR Petri nets. A deadlock in context of FMS means that during processing, some parts requests for some resources currently held by some other parts in this set. This leads to indefinite waiting and hence starvation. Some solutions to deadlock are deadlock prevention, deadlock avoidance, deadlock detection & recovery and deadlock ignorance. This paper focuses on deadlock avoidance approach by introducing a modified version of the Banker's Algorithm.

The authors explain the S^*PR petri net by stating an example of a production cell where two types of parts have to be processed. The cell has four machines, each of which can process 2 parts at a time and two tool stores which contain 4 classes of tools. Rules are specified as to which machine can use what tool for processing the parts. Apart from machines and tools, robots are also present to move parts between components of the cell and also to load & unload the machines from input & output of the cell. Buffers are used for intermediate storage of incompletely processed parts. The authors also present the formal definition of S^*PR Petri nets. The net is defined as $N = < P, T, C >$, where P is a partition (set of all places) such that $P = P_0 \cup P_S \cup P_R$. Places of P_S are process state places. A token in $p \in P_S$ models an in-process part. Places of P_0 are idle state places (state in which processes are idle). Places of P_R handle resources used by active processes. T is the set of all transition nodes. C is the incidence matrix of the petri net graph. It is assumed that the Petri net is strongly connected. Also, for each $r \in P_R$, there exists a unique P-invariant (Semiflow) Y_r, such that $Y_r(r) = 1$ (minimal P-semiflow). For a given $p \in P_S$, $Y_r(p) = k$ means that k copies of r are used by every token(process) in that state modelled by place p. It implies that resources can neither be created nor destroyed. These P-Semiflows enforce the resources to be reusable.

In Banker's Algorithm, we require the following: the maximum need of resources for each active process, dynamic information about resources assigned to each process and a set of available resources in order to determine if an ordering for sequential termination of processes exists. The algorithm verifies that a given system state is safe by the following procedure: Find an active process able to terminate using the resources it holds as well as the available resources. If no such process exists, the state is marked as unsafe, else withdraw the process from the set of active processes, release its held resources. If every process can be removed from the set of active processes, the state is safe. This approach gives a sufficient but not a necessary condition for safeness. Now, speaking in terms of petri net, a process is nothing but a token in state place. A process is said to be $M_R - terminable$ if there exists a path joining its state place and the corresponding idle state, in such a way that the path can be followed by the process using the held resources as well as the available resources. In other words, if there are enough resources for the process termination when rest of the active processes do not move from their current state. The authors propose a condition for a certain process to be $M_R - terminable$: $M_R + Y_R(\pi_M(a)) - Y_R(p) \geq 0$, where M_R is a reachable marking, Y_R is P-invariant vector, π_M is the mapping of active process to its corresponding state place, a is the active process and p is its state place. The authors propose a function **isBankerAdmissible** which returns a Boolean true if a sequential termination order of M active processes exists, else it returns false. The complexity of this function is $O(|P_S|^2 |P_R| \max(T_i))$ which is polynomial in Petri net model size. This function employs a helper function named, **isTerminable** which verifies whether an active process corresponding to a reachable marking M is $M_R - terminable$.

In conclusion, this paper has two main contributions. First one is introduction of a generalised class of Petri nets named S^*PR to model FMS. The second being modification of Banker's Algorithm for implementing deadlock avoidance in S^*PR nets in polynomial time. The authors remark that concept of "zone" or "control point" can be used to improve this approach. It might be sufficient to move the part to a state in which there is no interaction with rest of the parts.