We consider a finite, undirected graph each node of which contains a process. Processes contained in nodes directly connected by an edge of the graph are called each other's neighbours.

An act of communication is only possible between two neighbours. At any moment in time each process is ready to communicate with precisely one of its neighbours; the act of communication between two neighbours can only take place when each of them is ready to communicate with the other, and, as soon as they are both ready to communicate with the other, the communication is assumed to take place within a bounded period of time.

For each node there exists an (otherwise arbitrary) cyclic order of its neighbours, and the act of communication with one of its neighbours causes the node to become ready to communicate with its next neighbour, where "next" is to be understood in terms of that cyclic order. It is this rigid rule of the locally cyclic communication patterns that justifies the word "simple" in the title of this note. For such systems we shall determine the conditions characterizing the absence of the dangers of deadlock or starvation.

We represent the state of each process by the presence of one arrow from its node towards (the node of) the neighbour it is ready to communicate with: hence each node has always one outgoing arrow along one of the edges of the original undirected graph. In this representation, the act of communication between two neighbours takes place when they point to each other; the act of communication causes a "rotation" of both outgoing arrows. In this representation, the absence of deadlock is equivalent to the existence of at least one edge along which two arrows (in opposite directions) are present.

Let c be an arbitrary cycle of the undirected graph, in which neither a node nor an edge occurs more than once. (Such cycles contain at least 3
different nodes.) On this cycle we choose an arbitrary direction, which gives each node a "right-hand" neighbour and a "left-hand" neighbour in the cycle. Because such cycles contain at least 3 nodes, these two neighbours are different. For the outgoing arrow of a node $x$ of that cycle we define a "signature with respect to $c$": if it points to a node that, in the cyclic order associated with $x$, lies in the range from (and excluding) the left-hand neighbour of $x$ to (and including) the right-hand neighbour of $x$ we call the arrow positive; otherwise we call the arrow negative.

**Lemma 1.** No act of communication changes the truth-value of the predicate: the outgoing arrows of the nodes of the cycle $c$ have the same signature with respect to $c$.

**Proof.** The value of the predicate can only change when the signature of the outgoing arrow of a node of $c$ is changed. This can only happen at an act of communication with either its left-hand, or its right-hand neighbour in the cycle $c$. This is only possible when two communicating neighbours on the cycle had outgoing arrows of different signature. The act changes the signature of both arrows, so their signatures remain different from each other. In short: if the predicate is false it remains false in spite of the possibility of changing signatures, if it is true, it remains true because none of the signatures can change. (End of proof.)

**Lemma 2.** The existence of a cycle $c$ with outgoing arrows with the same signature causes local deadlock and, if the original graph is connected, total deadlock.

**Proof.** None of the outgoing arrows of the nodes of $c$ can have its signature changed, hence for each node of $c$ the number of acts of communication it can perform is bounded (by a bound lower than the number of its neighbours). By induction, the number of acts of communication of any node that is connected to $c$ via a finite path, is bounded. (End of proof.)

**Lemma 3.** In the case of total deadlock there is at least one cycle with all its outgoing arrows of the same signature.

**Proof.** Total deadlock means that no process has its outgoing arrow "matched" by an arrow in the opposite direction. Starting at any node, the step that consists of going from that node to the node its outgoing arrow points to can be repeated indefinitely. On a finite graph we must visit a node visited before, and hence a cyclic path (of at least 3 nodes) must exist: but that is a cycle with all its outgoing arrows of the same signature. (End of proof.)

Combining lemma's 2 and 3 we conclude our main