(N, M_0) shown in Fig. 3.2, $M_2 = (1\ 0\ 1)$ is the minimum marking to make t_0 enabled but is not coverable. Thus t_0 is dead. As another example, let us change the initial marking for the net shown in Fig. 3.4(h) to $M_0 = (2\ 0)$. For the transition t_1 in the net , the minimum enabling marking is $M_{min} = (0\ 1)$, which is coverable since there exists a marking $M_3 = (1\ 1) \geq M_{min} = (0\ 1)$ such that $M_3 \in R(M_0)$.

3.6 Persistence, and Structural & Behavioral Conflicts

A Petri net (N, M_0) is said to be *persistent* if, for any two enabled transitions, the firing of one transition will not disable the other. A transition in a persistent net, once it is enabled, will stay enabled until it fires. The notion of persistence is useful in the context of parallel program schemata [82] and speed independent asynchronous circuits [122, 126]. Persistency is closely related to conflict-free nets [180], and a safe persistent net can be transformed into a marked graph by duplicating some transitions and places [45]. Note that all marked graphs are persistent, but not all persistent nets are marked graphs. For example, the net shown in Fig. 3.4(c) is persistent, but it is not a marked graph.

Exercise 3.3. Which of the Petri nets shown in Fig.3.4 are persistent and which are non-persistent.

Answer: The two nets shown in Fig.3.4 (d) and (f) are non-persistent and the remaining six nets are persistent.

Two transitions are said to be in a *structural conflict* if they have a common input place. Two transitions in a structural conflict are said to be in a *behavioral conflict* if there is a reachable marking in which both are enabled but firing one disables the other. Thus a net is persistent iff it has no two transitions in a behavioral conflict, or a net is non-persistent iff it has a pair of transitions in a behavioral conflict.