# EXAMPLES

| Name | Description | Invariant |
|---|---|---|
| BasicOneQueueOneProcessor | The basic Petri net for modeling a service post with a queue | P1+Working=1 |
| BasicOneQueueOneProcessorCountInCountOut | Same as previous, with in and out counters | P1+Working=1<br>CountIn = Waiting+Working+CountOut |
| BasicOneQueueOneProcessorExitQueue | The basic Petri net for modeling a service post with a queue. We add a queue for exiting. We keep the counters | P1+Working=1<br>CountIn = Waiting+Working+WaitToExiit+Checking+CountCheck+CountNC |
| Taxis | Models a system for taxicabs. There are two lines for people: one for single person rides; one for two people rides. The taxi cab must drive to a gate where there is a line for begin checked before leaving the lot. | CountInTaxi = TxQ+ CountSi<br><br>CountEnterSh=ShQueue+2*ShQueue<br>CountTaxi=TxQ+Driving+<br>+ExitQueue+Checking+CountTxOut |
| 1q2p | One que serviced by two different processors | Working1+ReadyProc1=1<br>Working2 + ReadyProc2 =1 |
| BasicOneQueuuUndistProcs | One que serviced by two indistinguishable processors | P1+Working=2<br><br>CountIn=Working+Waiting+CountOut |
| ThreeWithTurns | One Queue Three processors, but they cannot work simultaneously as they need the same machine M. They have to work in this order P1, P2,P3, P1,P2,P3 … | P1+w1=1<br>P2 +w2=1<br>P3 + W3 = 1 |
| PriorityQueue | Bank with one teller and 2 queues: priority and general. The teller sequentially services two people and then one from the general queue. | P1+WP1+P2+WP2+P3+WG=1<br><br>CountInP=Priority+WP1+WP2+CountP<br><br>CountInG=Queue + WG + CountG |

| Name | Description | Invariant |
|---|---|---|
| TwoAtATimeWeights | Basic Petri net, but processor. Services two at a time. We use weights | CountIn =Queue+2*Working+CountOut +WaitForOther+Temp1+Temp2<br><br>Proc+Working = 1 |
| TwoAtATimeNoWeights | Basic Petri net, but processor. Services two at a time. | CountIn =Queue+2*Working+CountOut<br><br>Proc+Working = 1 |
| CheckNoCheck | Models a checkpoint int which the guard checks one, does not check the next, checks, does not check, … | Ready + Checking <= 1<br><br>CountEnter=Queue+Checking+ CountCheck+CountNoCheck<br><br>CountNocheck = Checking+CountCheck |
| Count | Siimulates a counter from 0 -999 | CountIn = Temp + P0 +10*P1 + 100*P2 |
| ThreeTasksTwoSim | Simmultates Procese that have to complete three tasks: A, B, and C. B and C mus tbe performed at the same time. The order can be A (BC), or (BC) A. P1 executes A, P2 executes B, P3 executes C. | P1+W1_1+W1_2 = 1<br><br>P2=P3<br><br>P3+W23_1+W23_2=1<br><br>CountIn = Queue+Q1_2+Q23_2+ W1_1+W1_2+W32_1+W32_2+ |