7 Using a GSPN for Performance Evaluation

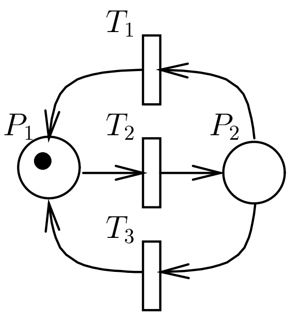
In this note we will consider two aspects of using a GSPN model of a system once it has been constructed: generating and solving a corresponding Markov process, and deriving performance measures. At the end of the note we will summarise the assumptions we are making when we use GSPN models.

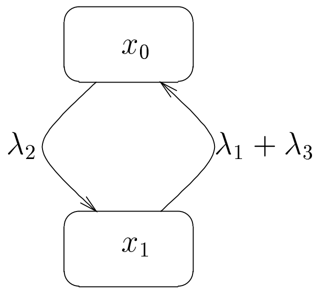
7.1 Generating and solving the corresponding Markov process

As stated in Section 5.2 generating the Markov process underlying an SPN model is very straightforward. We take advantage of the isomorphism between the reachability graph of the SPN and the state transition diagram of the Markov process. If the markings of the SPN are $\{M_0, M_1, \ldots, M_N\}$ (where M_0 is the initial marking), then the states of the Markov process will be $\{x_0, x_1, \ldots, x_N\}$ generated as follows:

- we associate a state, x_i, in the Markov process with every marking, M_i, in the reachability graph of the SPN;
- the transition rate from state x_i (corresponding to marking M_i) to state x_j (M_j), is obtained as the sum of the firing rates of the transitions that are enabled in M_i and whose firings generate marking M_j.

If we consider the very simple SPN model shown below, $M_0 = (1, 0)$ and $M_1 = (0, 1)$, are the only possible markings. Suppose that the firing rates of transitions T_1, T_2 and T_3 are λ_1, λ_2 and λ_3 respectively.





$$\mathbf{Q} = \begin{pmatrix} -\lambda_2 & \lambda_2 \\ \lambda_1 + \lambda_2 & -(\lambda_1 + \lambda_2) \end{pmatrix}$$

We associate states x_0 and x_1 with the markings M_0 and M_1. The transition rate from x_0 to x_1 is λ_2 because T_2 is the only transition enabled in M_0 whose firing results in M_1. The transition rate from x_1 to x_0 is $\lambda_1 + \lambda_3$ since both T_1 and T_3 are enabled in M_1, and the firing of either of them will result in the marking M_0.

Once the Markov process corresponding to an SPN model has been generated, it is solved in exactly the same way as Markov processes which are constructed directly during modelling. The steady state probability distribution is found by solving the global balance equations, together with the normalisation condition.

Two additional features were added to SPN notation to give GSPN notation: inhibitor arcs and immediate transitions. The effect of inhibitor arcs in a GSPN model is to alter the reachability graph of model: some markings and transition firings which would have been possible in the absence of the inhibitor arcs may no longer be possible. However, the effect of this new type of arcs only impacts on the generation of the reachability graph, not on the subsequent generation of the underlying Markov process. Once the reachability graph has been constructed obeying the constraints of the inhibitor arcs, they may be forgotten as far as generating the Markov process is concerned. Unfortunately, the same is not true for immediate transitions.

Recall that in a Markov process the delay spent in a state must be exponentially distributed. However, in a GSPN the delay in a vanishing marking is of length zero because the immediate transition will fire at once, moving the GSPN on to another marking. As far as the Markov process is concerned this means that these markings must be eliminated from the reachability graph before the state space of the Markov process is generated. Moreover, if immediate transitions from a marking can lead to two or more different markings, the transition rates to these markings need to be adjusted (by the decomposition principle), as explained below.

Let us consider first the simplest case: when a vanishing marking, M_v, enables a single immediate transition. In this case the next marking will clearly be the one resulting from firing the immediate transition. We will call this the *successor marking*, M_s. First we delete the vanishing marking from the reachability graph; similarly, we delete the arc *from* M_v *to* M_s. In principle, there may have been arcs *to* M_v *from* any other marking, M_i, in the reachability set. Each such arc is now taken directly from M_i to M_s; the name of the eliminated immediate transition is added to the label of the arc, but the rate or probability associated with the arc is unchanged.

If a vanishing marking, M_v enables more than one immediate transition it represents a conflict state, and there will be more than one arc leaving the marking in the reachability graph, each arc leading to a different possible successor marking. We will call these markings the *successor set*. As before, we delete M_v from the reachability graph. We also delete all arcs *from* M_v *to* markings in the successor set. For each arc which did come *to* M_v, we now form an arc to every marking in the successor set. Thus, if there are three markings in the successor set (i.e. the vanishing marking enabled three immediate transitions) then a single arc to the vanishing marking which has now been deleted will be replaced by three arcs, one to each of the three successor markings. As in the case of a single immediate transition, the name of the transition which has been removed is added to each of these new arcs. In addition, if these arcs represent timed transitions the rate associated with the arc must be adjusted in each case to represent the probability of that successor marking. For example, if the rate from some marking M_i to the vanishing marking, M_v, was r, and M_v enabled two immediate transitions, whose firing weights were equal (i.e. they were equally likely to fire) then the rate to each of the two successor markings will be $r/2$. Similarly, if the arcs represent immediate transitions then the relative probability of each of the new arcs must be chosen to reflect the original probability of the arc multiplied by the probability of the immediate transition which has been removed. For example, if the probability of the transition from some marking M_i to the vanishing marking, M_v, was p, and, as above, M_v enabled two immediate transitions whose firing weights were equal, then the probability of the transitions from M_i to the two successor markings would be $p/2$.

This procedure is systematically applied to all the vanishing markings in the reachability graph. In the end all arcs in the modified reachability graph will have a rate originating from a timed transition associated with every arc. As explained above, this rate may have been adjusted during the elimination of vanishing markings to reflect the relative probability of immediate transitions enabled after the timed transition. It is this modified reachability graph that is used to generate the Markov process underlying a GSPN model. Each tangible marking corresponds to a state in the Markov process, and the transition rate between states of the Markov process are derived as described above for SPNs. Once the Markov process is generated, it is solved as before, via the generator matrix.

7.1.1 Example: The reader-writer system revisited

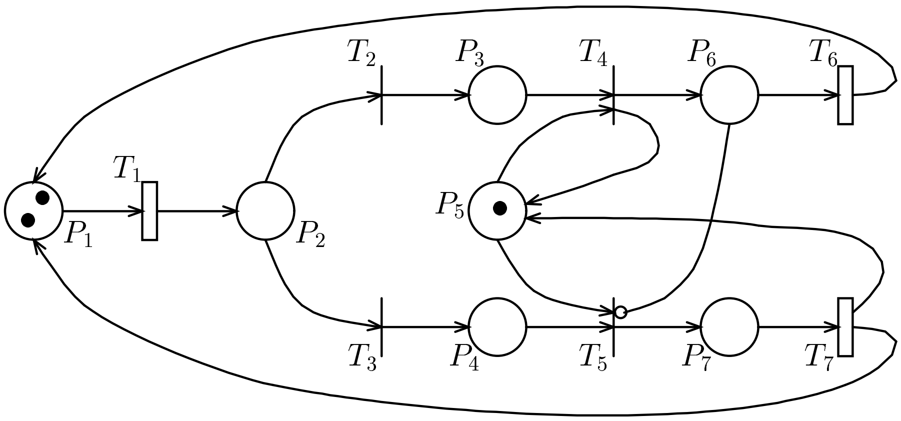


Figure 10: GSPN representing the simple reader-writer system

M_0	$(2, 0, 0, 0, 1, 0, 0)$	tangible
M_1	$(1, 1, 0, 0, 1, 0, 0)$	vanishing
M_2	$(1, 0, 1, 0, 1, 0, 0)$	vanishing
M_3	$(1, 0, 0, 1, 1, 0, 0)$	vanishing
M_4	$(1, 0, 0, 0, 1, 1, 0)$	tangible
M_5	$(1, 0, 0, 0, 0, 0, 1)$	tangible
M_6	$(0, 1, 0, 0, 1, 1, 0)$	vanishing
M_7	$(0, 0, 1, 0, 1, 1, 0)$	vanishing
M_8	$(0, 0, 0, 1, 1, 1, 0)$	tangible
M_9	$(0, 0, 0, 0, 1, 2, 0)$	tangible
M_{10}	$(0, 1, 0, 0, 0, 0, 1)$	vanishing
M_{11}	$(0, 0, 1, 0, 0, 0, 1)$	tangible
M_{12}	$(0, 0, 0, 1, 0, 0, 1)$	tangible

Table 1: Table showing the markings of the reader-writer GSPN model

We will now illustrate the elimination of vanishing markings for the reachability graph of the reader-writer GSPN model, shown in Figure 10. Note that the parametric marking of place P_1, considered in the previous lecture note, has now been replaced in the initial marking by two tokens in P_1.

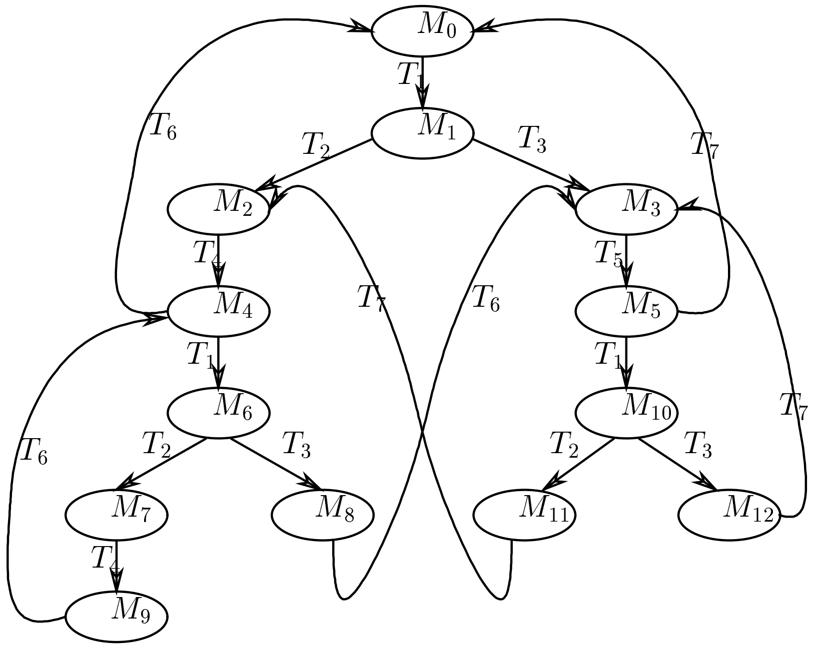


Figure 11: Reachability graph of the reader-writer GSPN model

The reachability set of the model is given in Table 1, together with an indication of whether each marking is vanishing or tangible, i.e. whether it enables an immediate transition or not. Note that there are 13 markings; if we considered the untimed Petri net with the same structure it would have 19 states. The reduction is caused by the priority of immediate transitions.

The full reachability graph, including vanishing markings, is shown in Figure 11.

Exercise: In this, and each of the modifications of the reachability graph, the transition rates have been omitted from the arcs—*you should write them in.* (Recall that the rate at which each process undertakes independent work is λ, the rate at which a process can perform a read access is r and the rate at which a process can perform a write access is w. The probability that a process chooses to read is 0.6, while the probability that it chooses to write is 0.4.)

In Figure 12 the effect of removing the vanishing markings which enable the choice between T_2 and T_3 is shown. For example, marking M_1 is now eliminated. The arc which previously went from M_0 to M_1 is now replaced by two arcs, one from M_0 to M_2 and one from M_0 to M_3. These arcs are labelled by the transitions which must fire to make this transformation of the marking occur: in this case, $T_1 + T_2$ and $T_1 + T_3$. The rates labelling arcs will also be adjusted. For example in M_0 the rate of the transition to M_1 will be $2 \times \lambda$ since independent processing takes place concurrently. The choice of transition T_2 occurs with probability 0.6. Therefore the rate which will be associated with the arc from M_0 to M_2 in the modified reachability graph is $1.2 \times \lambda$. Similarly, the arc from M_4 to M_7 will now be labelled with rate $0.6 \times \lambda$.

Once the vanishing markings which enabled the choice between T_2 and T_3 have been

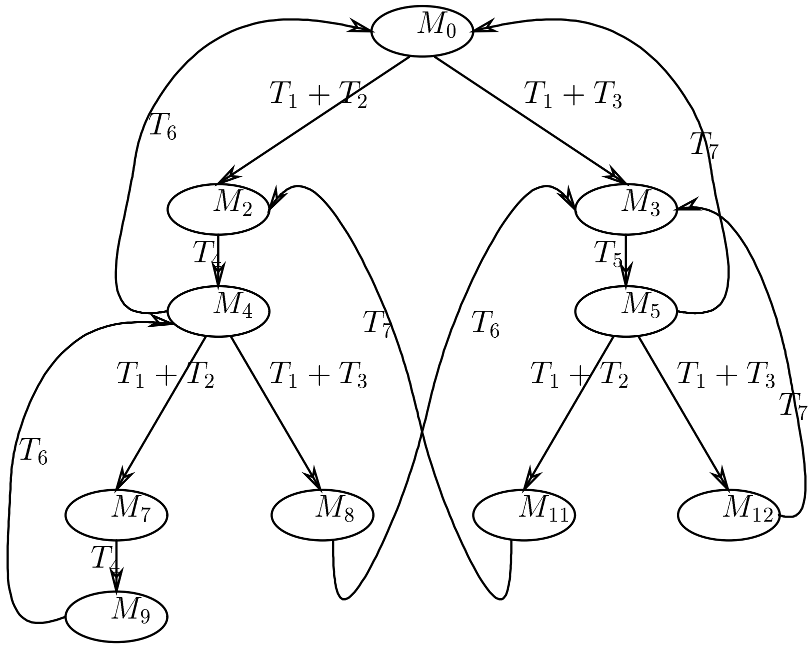


Figure 12: Reachability graph of the reader-writer model with vanishing markings which enable choices removed

eliminated, it remains to remove those markings which enable T_4 and T_5. Note that it is always the case that if one of these transitions is enabled in a marking it is the only transition enabled. Therefore elimination of the corresponding vanishing markings is simpler in this case. For example, M_2 enables T_4. In Figure 13 we can see that this marking has been deleted. The arc from M_0 to M_2 (introduced in the last step) is now deleted and replaced by an arc from M_0 to M_4. Similarly arcs from M_{11} and M_9 are now redirected to M_4. In each case the label of the arc is changed appropriately.

7.2 Deriving performance measures

The steady state probability distribution, $\boldsymbol{\pi}$, is still the basis of performance evaluation. In other words the aim is to derive performance characteristics of the system based on the steady state probability of being in certain states, or markings. Recall that when we were modelling directly at the Markov process level, defining performance measures, such as the average number of data packets waiting at a PC, involved examining the state representation of each state. Now, however, we can identify the states we are interested in by their characteristics at the GSPN level.

At the GSPN level the states (markings) which we are interested in can usually be identified either by whether a particular transition is enabled, or by whether a particular place is marked. To derive a performance measure we then associate a value with each of the markings we are interested in, just as we did when working directly at the Markov process level (cf. Section 3.3). For example, to derive the utilisation of the database in the reader-writer system, we associate a value 1 with any marking in which transitions T_6 or T_7 are enabled, and a value 0 with all other markings. In the CSPL file for SPNP

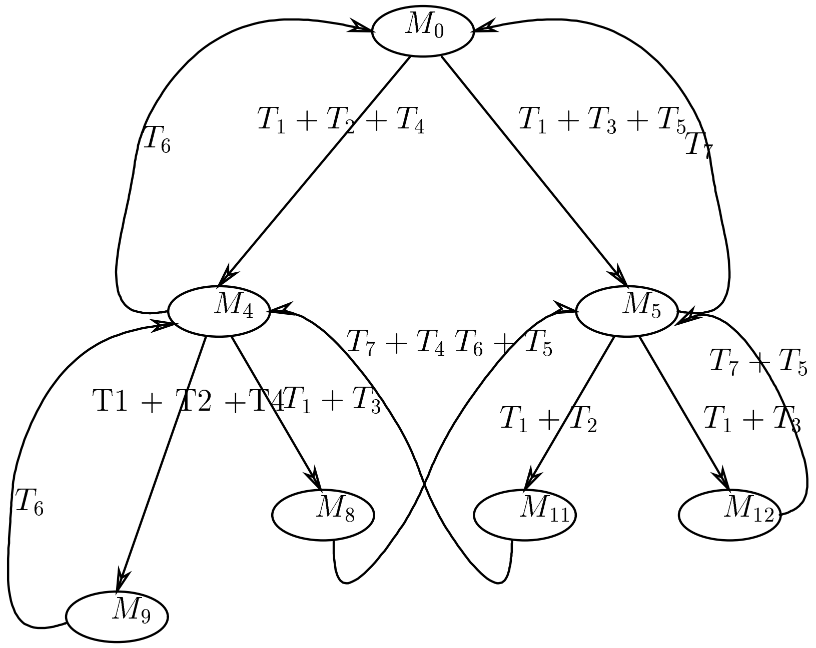


Figure 13: Reachability graph of the reader-writer model with all vanishing markings removed

this is expressed as:

```
 double util()
{ if (enabled("t6") || enabled("t7")) return(1.0); else return(0.0); }
```

The value associated with each marking, 1.0 or 0.0 in the case of `util`, is generally termed a *reward*. Different rewards can be used to calculate different measures. Indeed, typically a reward over all markings must be defined for each performance measure to be calculated. For example, if we wanted to derive the throughput of accesses to the database we associate a reward of $w(= 0.01)$ with all markings in which T_7 is enabled, and a reward of $r(= 0.05)$ multiplied by the number of readers (number of tokens in P_6) with all markings in which T_6 is enabled. In CSPL this is expressed as:

```
double tput() {
    if (enabled("t7"))
        return(0.01);
    else if (enabled("t6"))
        return(0.05*mark("p6"));
    else
        return(0.0);
}
```

The expected values of these rewards is generated and printed in the output file by the appropriate calls to `pr_expected` in the `ac_final` function:

```
void ac_final() {
```

```
    pr_mc_info();
    pr_std_average();

    pr_expected("Utilisation",util);
    pr_expected("Throughput",tput);
}
```

7.3 Assumptions

Since we are using the GSPN to generate a Markov process which we then solve numerically using the techniques discussed in lecture note 3 the assumptions we need to make about our model are the same ones as are needed for Markov processes in general. However, since we are now modelling at the GSPN level rather than the state space level, it is perhaps more natural to consider what these assumptions tell us about the GSPN.

Recall that in order to ensure that the stochastic process we were considering was a Markov process we made the assumption that all delays and inter-event times within the model were exponentially distributed. As we have seen, in a GSPN this condition is violated because some inter-event times have no duration: in a vanishing marking the time until the next event (transition firing) is zero. However, as we have demonstrated that it is possible to get round this problem by eliminating the vanishing markings before the Markov process is generated.

Assumptions we previously made to ensure that the steady state probability distribution coincided with the long-term probability distribution are that the model is finite, time homogeneous and irreducible. Since we still rely on the steady state probability distribution to derive performance measures we are making the same assumptions for GSPN.

Finite implies that the number of markings in the reachability set of a model (both tangible and vanishing markings) is finite. It can be shown that a GSPN is finite if it is *bounded*. A place in a Petri net is k-bounded if the number of tokens in the place will never exceed k. A Petri net is bounded if every place in the net is K-bounded for some finite value K.

Time homogeneity implies that the firing characteristics and system dynamics of a model do not change depending on when you observe it. This does not necessarily mean that these characteristics are static—we have seen that marking dependent rates and infinite server semantics may mean that the rate at which a transition fires may vary—but that firing rate will change only as dictated by the state of the model, not by how long it has been running.

Irreducibility implies that it is possible to reach an arbitrary state from every other state. In particular it must be possible to reach the initial marking from every reachable marking of the GSPN, and this implies that the reachability graph is strongly connected.

Jane Hillston ⟨jeh@inf.ed.ac.uk⟩. September 19, 2003.