Properties and Bounds on P/T Nets: An Example of Application[*]

Javier Campos
Universidad de Zaragoza, Spain

Let us present an example of application for the computation of bounds in the case of the Timed Well-Formed Coloured Net (TWN) model of a shared-memory multiprocessor depicted in Figure 1. The architecture comprises a set of processing modules interconnected by a common

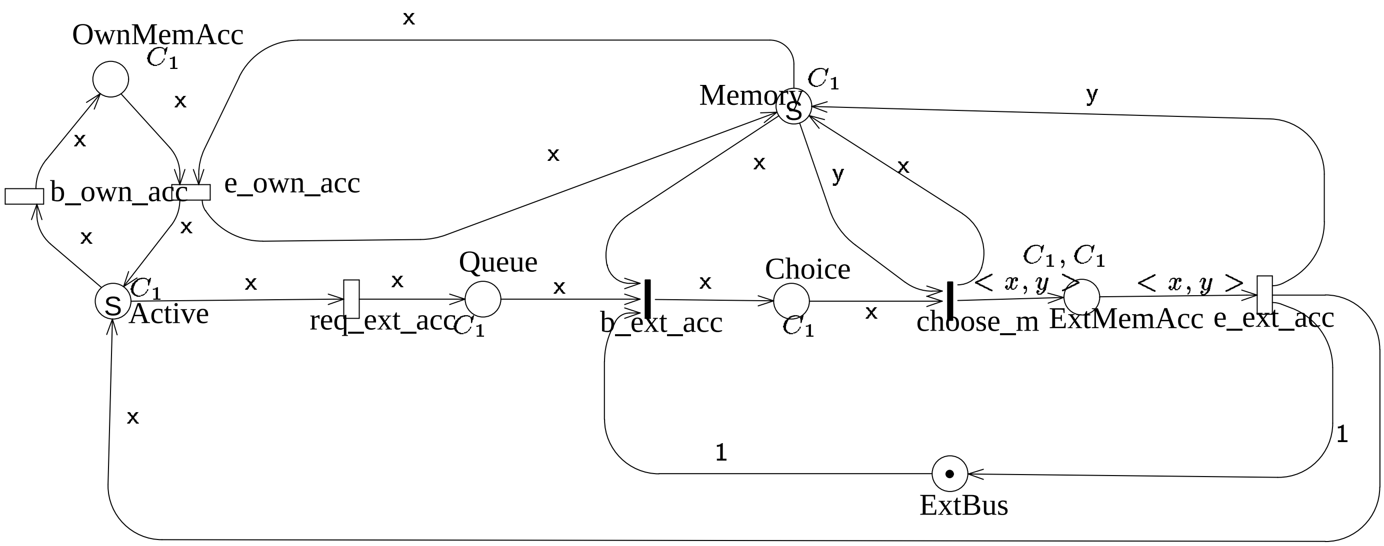

Figure 1: TWN model of a shared-memory multiprocessor.

bus called the "external bus". A processor can access its own memory module directly from its private bus through one port, or it can access non-local shared-memory modules by means of the external bus. In case of contention for the access to one shared-memory module, preemptive priority is given to external access through the external bus with respect to the accesses from the local processor. The experiments on the shared-memory model have been carried out assuming to have 4 processors and that the average service time of all the transitions are equal to 0.5.

According to the arguments presented in the Tutorial, bounds can be computed solving linear programming problems with constraints included in Table 1, where the first letters of each

[*]Tutorial of PNPM'99–PAPM'99–NSMC'99, Zaragoza (Spain), September 6-10, 1999. This text has been extracted from the paper "Operational analysis of timed Petri nets and application to the computation of performance bounds", by G. Chiola, C. Anglano, J. Campos, J.M. Colom, and M. Silva, in Proceedings of the *5th International Workshop on Petri Nets and Performance Models*, pp. 128-137, Toulouse, France, October 1993, IEEE Computer Society Press.

$$
\begin{array}{ll}
(c_1) & \overline{\boldsymbol{\mu}}[Active] = 4 + \boldsymbol{\sigma}[e\_e\_a] + \boldsymbol{\sigma}[e\_o\_a] - \boldsymbol{\sigma}[r\_e\_a] - \boldsymbol{\sigma}[b\_o\_a]; \\
 & \overline{\boldsymbol{\mu}}[Memory] = 4 + \boldsymbol{\sigma}[e\_e\_a] - \boldsymbol{\sigma}[b\_e\_a]; \\
 & \overline{\boldsymbol{\mu}}[OwnMemAcc] = \boldsymbol{\sigma}[b\_o\_a] - \boldsymbol{\sigma}[e\_o\_a]; \\
 & \overline{\boldsymbol{\mu}}[Queue] = \boldsymbol{\sigma}[r\_e\_a] - \boldsymbol{\sigma}[b\_e\_a]; \\
 & \overline{\boldsymbol{\mu}}[Choice] = \boldsymbol{\sigma}[b\_e\_a] - \boldsymbol{\sigma}[c\_m]; \\
 & \overline{\boldsymbol{\mu}}[ExtMemAcc] = \boldsymbol{\sigma}[c\_m] - \boldsymbol{\sigma}[e\_e\_a]; \\
 & \overline{\boldsymbol{\mu}}[ExtBus] = 1 + \boldsymbol{\sigma}[e\_e\_a] - \boldsymbol{\sigma}[b\_e\_a]; \\
(c_2') & \boldsymbol{\chi}[e\_e\_a] + \boldsymbol{\chi}[e\_o\_a] = \boldsymbol{\chi}[r\_e\_a] + \boldsymbol{\chi}[b\_o\_a]; \\
 & \boldsymbol{\chi}[b\_e\_a] = \boldsymbol{\chi}[c\_m] = \boldsymbol{\chi}[e\_e\_a] = \boldsymbol{\chi}[r\_e\_a]; \\
(c_3) & \boldsymbol{\chi}[b\_o\_a] = \boldsymbol{\chi}[r\_e\_a]; \\
(c_4\&c_5) & \boldsymbol{\chi}[b\_o\_a]\ \overline{\mathbf{s}}[b\_o\_a] = \overline{\boldsymbol{\mu}}[Active]/2; \\
 & \boldsymbol{\chi}[r\_e\_a]\ \overline{\mathbf{s}}[r\_e\_a] = \overline{\boldsymbol{\mu}}[Active]/2; \\
 & \boldsymbol{\chi}[e\_e\_a]\ \overline{\mathbf{s}}[e\_e\_a] = \overline{\boldsymbol{\mu}}[ExtMemAcc]; \\
(c_4) & \boldsymbol{\chi}[e\_o\_a]\ \overline{\mathbf{s}}[e\_o\_a] \le \overline{\boldsymbol{\mu}}[OwnMemAcc]; \\
 & \boldsymbol{\chi}[e\_o\_a]\ \overline{\mathbf{s}}[e\_o\_a] \le \overline{\boldsymbol{\mu}}[Memory]; \\
(c_6) & \boldsymbol{\chi}[e\_o\_a]\ \overline{\mathbf{s}}[e\_o\_a] \ge \overline{\boldsymbol{\mu}}[OwnMemAcc] + \frac{\mathbf{b}[OwnMemAcc]}{\mathbf{b}[Memory]}\overline{\boldsymbol{\mu}}[Memory] \\
 & \qquad -\mathbf{b}[Memory]; \\
(c_7) & 4\left(\overline{\boldsymbol{\mu}}[ExtBus] - \mathbf{b}[ExtBus]\left(1 - \frac{\overline{\boldsymbol{\mu}}[Memory]}{\mathbf{b}[Memory]}\right)\right) \le 0; \\
 & 4\left(\overline{\boldsymbol{\mu}}[ExtBus] - \mathbf{b}[ExtBus]\left(1 - \frac{\overline{\boldsymbol{\mu}}[Queue]}{\mathbf{b}[Queue]}\right)\right) \le 0;
\end{array}
$$

Table 1: Constraints for the model in Figure 1.

transition name have been used for reasons of space. The solution for the linear programming problems leads to upper and lower bounds, for the throughput of transitions, given by

$$\frac{8}{11} \le \boldsymbol{\chi}[e\_e\_a] \le 2$$

while the "exact" solution with exponential distribution is

$$\boldsymbol{\chi}[e\_e\_a] = 1.71999$$

An improvement in the lower bound can be obtained observing that when a token arrives in place Choice transition choose_m is enabled at least for one transition instance. This implies that the average marking of place Choice is equal to 0 (transition choose_m is immediate), so

$$\overline{\boldsymbol{\mu}}[Choice] = 0; \quad \mathbf{b}[Choice] = 0$$

(only tangible markings are considered) can be added to the set of constraints. Moreover place Memory is implicit with respect to the enabling of transition b_ext_acc, so we can consider this transition as having only two input places, so constraint (c_6) can be applied instead of constraint (c_7). Finally,

$$\mathbf{b}[Queue] = 3$$

can be added since the output transition of place Queue is immediate, and from the behaviour of the model it is clear that at most 3 processors can be waiting in the queue. The relations (c_7) in the above linear programming problem can thus be replaced with the new constraint:

$$4\left(\overline{\boldsymbol{\mu}}[ExtBus] + \frac{\mathbf{b}[ExtBus]}{\mathbf{b}[Queue]}\overline{\boldsymbol{\mu}}[Queue] - \mathbf{b}[ExtBus]\right) \leq 0$$

where $\mathbf{b}[Queue] = 3$. Solving this reduced linear programming problem the values obtained for the upper and lower bounds are:

$$1 \leq \boldsymbol{\chi}[e\_e\_a] \leq 2$$