I Introduction and Motivations.

One way of using Petri nets for modelling systems is to represent states by markings, and actions that modify the states by transitions. In order to represent the fact that one event can cause different modifications of the system, one has to label the transitions. For example, in a school, the beginning and the end of classes are signalled, or labelled, by the ringing of a bell. One possible way of investigating the functioning of a system is to examine the sequences of labels corresponding to the firing sequences of transitions of the Petri net that models the system. Several classes of languages have been defined and studied: languages of sequences of firing of transitions when one event or signal causes one action, and languages of words labelling firing sequences of transitions when one event can cause several actions.

In several cases, especially in the case of industrial systems, an intermediate situation occurs: one event can cause different actions, but given the state of the system, one event can cause only one action. This can be seen on the Petri net modelling the system by the fact that, given a reachable marking $M$ and a label $l$, there is at most one transition labelled by $l$ that is enabled. An example is given in Figure 1. We will call labelled Petri nets having this property deterministic labelled Petri nets.

In this paper, we shall give some properties of the languages of deterministic labelled Petri nets, and compare the classes of languages associated with them to the classes previously defined. As will be seen, the classes we present in this paper are intermediate in terms of power of expression with classes previously defined.
II Definitions.

We will consider only Petri nets with infinite token capacity. The following definitions follow:

A labelled Petri net $N$ is a triplet $(PN, \Sigma, h)$ where
- $PN$ is a Petri net with set of transitions $T$ and initial marking $M_0$
- $\Sigma$ is a finite set of labels
- $h$ is the labelling function from $T$ to $\Sigma \cup \{\lambda\}$, $\lambda$ being the empty word. The function $h$ is extended to a homomorphism of $T^*$ to $\Sigma^*$

The usual relation $M[t\rightarrow]M'$ is extended to words of $T^*$ by $M[w] M'$ if $\exists M'' \rightarrow M'$ and $M''[w] M$.

$F(N)$ the language of firing sequences is defined by:

$$F(N) = \{ w \in T^* \mid M_0[w] M \}$$

Given a finite set of terminal markings $\mathcal{M}$ the language of terminal firing sequences $F_0(N, \mathcal{M})$ is then:

$$F_0(N, \mathcal{M}) = \{ w \in T^* \mid M_0[w] M, \text{ and } M \in \mathcal{M} \}$$

$L(N)$, and $L_0(N, \mathcal{M})$ are defined by: $L(N) = h(F(N))$, $L_0(N, \mathcal{M}) = h(F_0(N, \mathcal{M}))$

In $1$ $M_0 \in \mathcal{M}$, while in $3$ there is no such restriction. This difference will not influence our results.

The classes of languages $L_1, L_2, L_3, L_4$ are then defined by

$$L_1 = \{ L \mid \exists N, L = F(N) \}$$

$$L_2 = \{ L \mid \exists N, L = F_0(N, \mathcal{M}) \}$$

$L_3$ and $L_4$ are obtained from $L_1$ and $L_2$ by adding the condition that no transition is labelled by $\lambda$ (the labelling function is continuous).

We will say that a labelled Petri net $N = (PN, \Sigma, h)$ is deterministic when:

$$\forall M \in M_0[-\gamma], \forall 1 \in \Sigma, \text{ there is at most one transition } t \text{ labelled by } 1 \text{ that is enabled.}$$

$N$ is strictly deterministic when it is deterministic and it satisfies the condition:

$$\forall M \in M_0[-\gamma], \forall t' \text{ labelled by } \lambda, \text{ it is not enabled whenever a transition } t \text{ not labelled by } \lambda \text{ is enabled.}$$

This last condition is used for modelling systems where certain actions can be performed only when no other action is possible (like a call for the operator). Figures 1 and 2 give examples of strictly deterministic Petri nets.

In general, the property of being deterministic depends on the labelling function and on the initial marking. In example 2, however, it does not depend on the labelling function.