A SURVEY ON THE DECIDABILITY QUESTIONS FOR CLASSES OF FIFO NETS

A. Finkel* and L. Rosier**

* Centre de Recherche Informatique de Montréal and University of Montréal (1987-1988)
1550 bd de Maisonneuve Ouest, Bureau 1000, Montréal (Québec), Canada H3G 1N2.
University of Paris-Sud, L.R.I., Bât. 490, 91405 Orsay Cedex, France (permanent address).
This work was supported in part by the Programme d'Action Structurante du Ministère de l' Enseignement Supérieur et de la Science du Québec.

** University of Texas at Austin, Dept. of Comp. Sci., Austin, TX 78712-1188, USA.
This work was supported in part by U.S. Office of Naval Research Grant No. N00014-86-K-0763 and National Science Foundation Grant No. CCR - 8711579.

Content.

1. Introduction
2. Definitions and general fifo nets
3. Monogeneous fifo nets
4. Linear fifo nets
5. Topologically free choice fifo nets
6. Conclusion
7. Bibliography

1. Introduction.

Over the years a number of automata theoretic models for parallel computation have enjoyed the attention of researchers. Petri nets (PNs) appear to be the most utilized such model; although systems of communicating finite state machines (CFSM's) are very popular among researchers concerned with communication protocols. See e.g. [Petri 86], [Reisig 85], [Brauer... 86a], [Berthelot 82], [Brams 83], [Zhao... 86], [Bochmann... 87], [Sarikaya... 86]. Recently many papers [Flé... 85], [Finkel 86], [Finkel...87], [Choquet...87] have considered the new automata theoretical model of parallel computation proposed by Memmi and Martin [Martin... 81] -- the Fifo Net (FN).

FNs are a generalization of both PNs and systems of CFSM's, a marriage that utilizes the transition mechanism of PNs in conjunction with the fifo channels of systems of CFSM's. Transitions in PNs add/substract a specific number of tokens to/from each place in the net. A place in a PN then behaves like a counter. Transitions in FNs enqueue/dequeue words (i.e., strings of symbols) to/from each fifo in the net. A fifo in a FN then behaves like a queue. Since the introduction of FNs in [Martin...81] many variations and restrictions of the basic model have been considered. cf. [Finkel 82a], [Starke 83], [Finkel 86], [Choquet...87], [Finkel... 87]. In this survey, we attempt to categorize the main decidability results concerning FNs.
This paper should be compared and contrasted with the earlier survey on FNs by Roucairol [Roucairol 86]. Although Roucairol's paper discusses some decidability issues, the reader should find our paper considerably more comprehensive in this regard.

One of the major reasons people model parallel systems using automata theoretical techniques is to provide a formal framework in which to analyze the systems. For example, with respect to communication protocols the analysis problems concerning the detection of various design errors, e.g. unboundedness, deadlocks, unspecified receptions, etc., are of paramount importance. See e.g. [Brand... 83], [Zhao...86]. Generally it is the case that the detection of design errors with respect to systems corresponds directly to solving one or more of the classical analysis problems with respect to the automata theoretic model.

There are many reasons that PNs are often the model of choice. One reason is that PNs have been around awhile and a great deal is known about them. See e.g. [Brauer... 83], [Reisig 85], [Peterson 81]. Another, perhaps more important reason, is that most of the classical analysis problems are decidable with respect to PNs. (This is not true for systems of CFSM's and FNs.) However, there are cases where modelling via PNs or systems of CFSM's becomes somewhat cumbersome. For example, in many parallel or distributed systems, processes communicate by sending messages to and from one another via fifo channels. PNs do not possess a built-in mechanism for directly modelling this type of behavior. Systems of CFSM's do possess such a mechanism and are frequently used to model systems of this sort. (This is one of the reasons that systems of CFSM's are often utilized in the modelling and analysis of communication protocols [Bochmann...78], [Aho...79], [Bochmann... 87], [Sarikaya... 86].) But while systems of CFSM's employ an implicit mechanism for handling queues they severely constrain the computational power of the independent processes as well as the way in which the processes communicate. These constraints might prove cumbersome when trying to model, for example, portions of an operating system where the various queues tend to contain messages from more than one process. FNs provide a more general structure that is able to model both of these examples in a straight-forward fashion.

There is a price to be paid for choosing either systems of CFSM's or FNs as a modelling tool. It is reasonably easy to show that both simple systems of CFSM's and simple FNs can simulate Post machines ; thus illustrating that the interesting analysis problems with respect to these systems are, in general, undecidable [Finkel 82], [Brand... 83], [Memmi 83], [Gouda...85], [Memmi... 85]. As a result a tradeoff between flexibility and generality must be sought. One wants restricted classes of these systems, that leave the fifo mechanism intact (in some important way), where most if not all of the analysis problems are decidable. Typical restrictions that have been considered involve restricting the allowable sequences of messages that can pass through a channel or fifo. The goal is to obtain better tools for the analysis of systems that contain some of the standard built-in data structures such as queues.

There are three basic restricted classes of FNs that have been considered in the literature -- Monogeneous FNs (MFNs), Linear FNs (LFNs), and Topologically Free Choice FNs (TFCFNs). In the remainder of this paper we offer a survey of what we feel are the most important decidability results with respect to each of these classes. Section 2 contains the general overall definitions concerning FNs utilized throughout the paper ; while Section 3 (4, 5, respectively) discusses the results concerning MFNs (LFNs, TFCFNs, respectively). We conclude in Section 6 with some general comments and provide a table that hopefully puts all of the results into perspective.

2. Definitions and general Fifo Nets - System of Communicating Finite State Machines.

In this section, we provide a general assortment of definitions concerning FNs. (Definitions utilized solely within a particular section are postponed until later.) Eight problems that will be considered throughout the paper are also defined. Lastly the computational power of general FNs is discussed as well as their relationship to systems of CFSM's.

2.1. Definitions.

FNs are like PNs except that places contain words or strings of characters rather than a number of tokens. The I/O protocol of such a place is like that of a fifo queue -- hence it is referred to as a fifo. More formally we have :

Definition 2.1. [Reisig 85] A *PN* is a tuple $H=(N,M_0)$ where $N = (F,T,V)$. F is a finite set of places, T a finite set of transitions, and $V : F{\times}T \cup T{\times}F \dashrightarrow \mathbf{N}$ the valuation function. $M_0 : F \dashrightarrow \mathbf{N}$ is the initial marking. A transition t is *fireable* from the marking M, written M(t>, if for every place f of F, $V(f,t) < M(f)$ where < denotes the usual ordering on **N**. When t is fired, the marking M' is reached where M' satisfies the following equation :

$$M'(f) = M(f) + V(t,f) - V(f,t) \qquad \forall f \in F.$$

We then write M(t>M' . Clearly the notion can be extended in the obvious way to sequences of transitions in T*. (end of Definition 2.1.)

The particular notations used here are nonstandard and are used in order to be consistent with the standard notations used for FNs.

Definition 2.2. [Martin...81] A *FN* is a tuple $H=(N,M_0)$ where N = (F,T,A,V) . F is a finite set of fifos, T a finite set of transitions, A a finite alphabet, and V : FxT U TxF --> A* the valuation function. M_0 : F --> A* is the initial marking. A transition t is *fireable* from the marking M, written M(t>, if for every fifo f of F, V(f,t) < M(f) where < denotes the Left Factor relation (i.e., x<y iff there exists z such that x.z=y). When t is fired, the marking M' is reached where M' satisfies the following equation:

$$V(f,t) \,.\, M'(f) = M(f) \,.\, V(t,f) \qquad \forall f \in F$$

We then write M(t>M' . Clearly the notion can be extended in the obvious way to sequences of transitions in T*.(end of Definition 2.2.)

The astute reader can now see that PNs are essentially FNs where |A| =1.
An example of a FN is shown in Figure 2.1. It models a communication protocol.

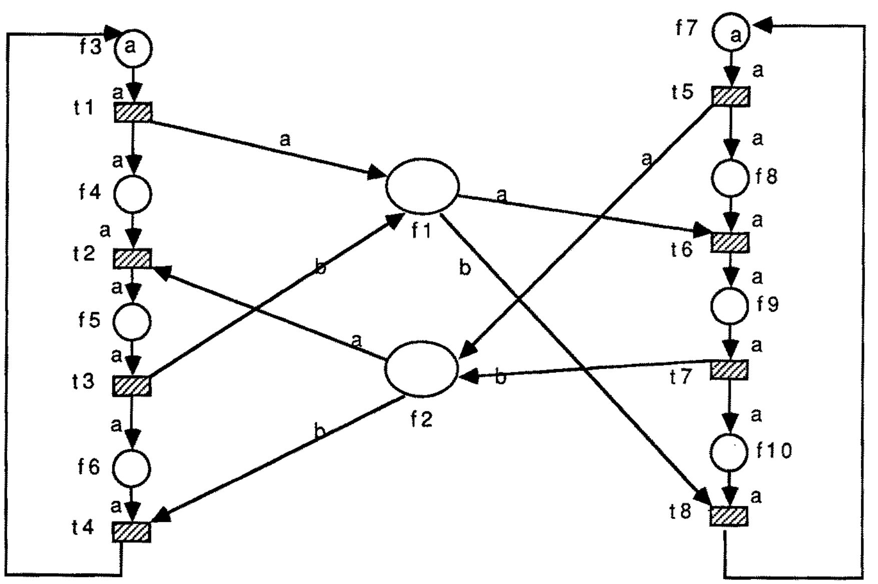

Figure 2.1.

For this example, $F = \{ f_1,...,f_{10} \}$, $T = \{ t_1,...,t_8 \}$, and $A = \{a,b\}$. The valuation function V should be clear from the diagram. $M_0 = (-,-,a,-,-,-,a,-,-,-)$ indicating that each of the fifos, except for f_3 and f_7, are initially empty. Fifos f_3 and f_7 both initially contain the string "a". Clearly, t_1 is fireable from M_0. In fact, $t_1t_5t_6t_2$ is a possible sequence of transitions :
$M_0(t_1> M(t_5> M'(t_6>M"(t_2>M'"$ etc... with M=(a,-,-,a,-,-,a,-,-,-); M'= (a,a,-,a,-,-,-,a,-,-); M"=(-,a,-,a,-,-,-,-,a,-); M'"=(-,-,-,-,a,-,-,-,a,-).

Definition 2.3. Let $R(N,M_0)$ denote the set of all markings that are *reachable* from M_0, i.e., $R(N,M_0) = \{ M \mid$ there exists x in T* such that $M_0(x>M \}$. Let $L(N,M_0)$ denote the *language* of the net or the set of all sequences in T* that are fireable from M_0, i.e., $L(N,M_0) = \{ x \mid x \in T^*$ and $M_0(x> \}$. An element $x \in T^*$ is said to be in the *center* of (N,M_0), denoted by $C(N,M_0)$, iff $M_0(x>M$ and $L(N,M)$ is infinite.

We are now in a position to define the problems considered in this summary. They are as follows :

1. **The Total Deadlock Problem (TDP)** : Given a FN (N,M_0) : Is $L(N,M_0)$ finite ?

2. **The Partial Deadlock Problem (PDP)** : Given a FN (N,M_0) : Is there a finite path in (N,M_0) that cannot be extended ? (I.e., is there an $x \in T^*$ such that $M_0(x>M$ where no transition in T is fireable from M ?)

3. **The Boundedness Problem (BP)** : Given a FN (N,M_0) : Is $R(N,M_0)$ finite ?

4. **The Reachability Problem (RP)** : Given a FN (N,M_0) and a marking M : Is $M \in R(N,M_0)$?

5. **The Quasi-Liveness Problem (QLP)** : Given a FN (N,M_0) : $\forall t \in T$, is there an $x \in T^*$ such that $M_0(xt>$?

6. **The Liveness Problem (LP)** : Given a FN (N,M_0) : $\forall M \in R(N,M_0)$, $\forall t \in T$, is there an $x \in T^*$ such that $M(xt>$?

7. **The Center Problem (CP)** : Is there an algorithm that will generate a recursive representation of $C(N,M_0)$?

8. **The Regularity Problem (Reg P)** : Given a FN (N,M_0) : Is $L(N,M_0)$ regular ?

In what follows we provide notations that are used throughout the paper.

Definition 2.4. We denote by $\Gamma(u)$ and by $\Gamma^{-}(u)$ the following sets : $\Gamma(u) = \{v \mid v \in T \cup F \text{ and } V(u,v) \neq \lambda\}$. $\Gamma^{-}(u) = \{v \mid v \in T \cup F \text{ and } V(v,u) \neq \lambda\}$. The *input alphabet* of a fifo is the set of symbols that appear in the valuation of at least one input arc of the fifo. The *output alphabet* is the set of symbols that appear in the valuation of at least one output arc of the fifo. The *alphabet* of a fifo f, denoted by A_f, is the union of the input and output alphabets for f. In subsequent sections, we consider three proper subclasses of FNs -- MFNs, TFCFNs, and LFNs -- two of which are defined by restricting the input language of the fifos. The *input language* of a fifo f in (N,M_0) is : $L_I(N,M_0,f) = h_f(L(N,M_0))$, where h_f is the homomorphism defined by $h_f(t) = V(t,f)$. The *output language* of a fifo f in (N,M_0) is : $L_O(N,M_0,f) = k_f(L(N,M_0))$, where k_f is the homomorphism defined by $k_f(t) = V(f,t)$.

With respect to a FN (N,M_0), we often associate a PN, (N_p,M_{0p}), called the *associated PN* and a coloured PN, (N_c,M_{0c}), called the *associated coloured PN.* Each has the same topology as (N,M_0). The only difference is in the valuation function. For the underlying PN, the valuations (with respect to (N,M_0)) for each arc are replaced by their respective lengths. For the associated coloured PN the valuations (with respect to (N,M_0)) are replaced by the multiset of alphabet symbols present in their respective words (i.e., each word is replaced by its Parikh image). Formal definitions follow.

Definition 2.5. [Finkel 86] Let (N, M_0) be a FN where $N=(F, T, A, V)$. The *associated Petri net* (N_p, M_{0p}) is defined as follows: $N_p = (F, T, V_p)$ with $V_p(x,y) = |V(x, y)|$ for (x, y) in $F \times T \cup T \times F$; and $M_{0p}(f) = |M_0(f)|$ for $f \in F$.

Definition 2.6. [Finkel 86] Let (N, M_0) be a FN where $N=(F, T, A, V)$. The *associated coloured PN* is defined by (N_c, M_{0c}) with $N_c = (F, T, C, V_c)$ where the colour function C associates with each fifo f its alphabet A_f. For every $(f, y) \in F \times T$, $V_c(f, y)$ is the multiset
$\{a\#(V(f,y),a) \mid a \in A_f\}$. For every $(x,f) \in T \times F$, $V_c(x,f)$ is the multiset $\{a\#(V(x,f),a) \mid a \in A_f\}$.
$M_{0c}(f)$ is the multiset $\{a\#(M_0(f),a) \mid a \in A_f\}$. The firing rule is $M(t > M'$ iff $\forall f \in \Gamma^{-}(t)$, $M(f) \geq V_c(f, t)$ and $M'(f) = M(f) + V_c(t, f) - V_c(f, t)$. (Here $\#(w,a)$ is the number of as in w.)

Let us remark that coloured PNs are defined in [Jensen 86].

2.2. The computational power of Fifo Nets.

The following theorem was shown in [Finkel 82b], [Memmi 83], [Memmi ... 85] :

Theorem 2.1. Given a Turing machine TM and an input x, one can effectively construct a FN that will simulate the computation of TM on x.

The theorem holds even for FNs whose edges are labelled by at most one letter (i.e., the range of the valuation function is contained within A U {λ}). The proof simply involves the observation that FNs can simulate Post machines. A simular result was stated for systems of CFSM's in [Brand ... 83]. See also [Gouda...85]. A consequence is stated in the following corollary. See [Finkel 82b], [Memmi 83], and [Memmi...85].

Corollary 2.2. There does not exist an algorithm for FNs with respect to any of the aforementioned eight problems.

2.3. Systems of Communicating Finite State Machines.

We present now the CFSM's model because the results (about FNs) in sections 3,4 allow to generalize some of the results in [Rosier... 84], [Gouda...85] obtained in the formalism of CFSM's. Another formalism used to model parallel systems is constituted by systems of CFSM's. Systems of CFSM's are often used for the modelling [Sunshine 81], analysis [Bochmann 78], [Brand...83], [Gouda...85], [Rosier...84] and synthesis [Aho...79], [Zafiropoulo...80] of communication protocols and distributed systems. In a system of CFSM's, the finite state machines communicate exclusively by exchanging messages via connecting channels. There are generally two one-directional, FIFO channels between each pair of machines in the system. Each machine has a finite number of states and state transition rules ; each state transition rule is accompanied by either sending or receiving one message to or from one of the machines output or input channels.

Definition 2.7. A *Communicating Finite State Machine* (CFSM) M is a labelled directed graph with two types of edges, sending and receiving. A sending (receiving) edge is labelled -a (+a) for some message a in a finite alphabet A of messages. One of the nodes in M is identified as the initial node. Let M and N be two CFSM's with the same alphabet ; the pair (M,N) is called a *system of two CFSM's*.

For example, a system of two CFSM's is portrayed in Figure 2.2. The two machines are M_1 and M_2. The transitions labelled -a, -b indicate that the transition is accompanied by sending an "a", "b" respectively, to the machines output channel. (Channel destinations are not explicitly given here as there is only a single input and output channel for each machine.) The labels +a and +b indicate that the message "a", "b" respectively is to be received. The starting state for M_1 (and M_2) is the state labelled 1. This example was taken from [Rosier ... 84] where a more thorough description of its behavior can be found.

It is reasonably easy to show that systems of CFSM's can be directly simulated by FNs. This is formalized in [Finkel 86]. As an example, consider again the system of CFSM's shown in Figure 2.2.

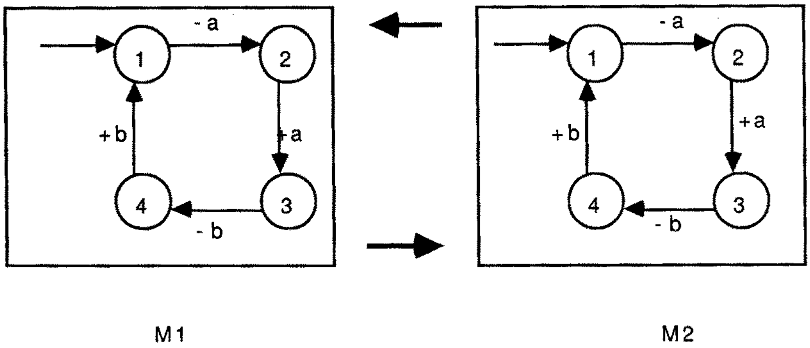

Figure 2.2.

The FN shown in Figure 2.1 directly simulates the system portrayed in Figure 2.2 . The fifo f_1 (f_2) is used to model the channel from M_1 to M_2 (M_2 to M_1) . Fifos f_3 through f_6 (f_7 through f_{10}) are used to simulate the finite state control of M_1 (M_2). Note that fifos f_3 - f_{10} behave like places in a PN since each can only receive a single character. The firing of transitions t_1, t_2, t_3, t_4 (t_5,t_6,t_7,t_8, respectively) corresponds to the transitions between states 1 and 2, 2 and 3, 3 and 4, 4 and 1 in machine M_1 (1 and 2, 2 and 3, 3 and 4, 4 and 1 in machine M_2). From this example, the astute reader should see how to construct a FN that directly simulates any given system of CFSM's.

3. Monogeneous Fifo Nets.

In the next three sections, we consider classes of FNs that admit algorithms for some of the eight problems defined in Section 2. Recall that there is no algorithm for any of the eight problems even for the class of FNs whose fifo alphabets are of size at most two. Now, PNs are essentially FNs whose fifo alphabets are of size one. All eight problems admit algorithms with respect to this class of FNs (i.e., with respect to PNs) [Finkel 82a], [Finkel 83], [Finkel 86], [Starke 83]. FNs whose fifos are bounded are equal to finite automata. Again all eight problems admit algorithms with respect to this class. Algorithms do not exist for classes that allow one unbounded fifo over a two letter alphabet.

In this section we discuss MFNs and some of their properties. For systems of CFSM's it was shown in [Rosier...84] [Rosier...86] that the BP was decidable if one of the machines was restricted to send only a single type of message -- i.e., the message alphabet for one of the two fifos was a single character. Now a PN is essentially a FN whose fifo alphabets are restricted to single characters. MFNs restrict the language of each fifo to be included in the finite union of LeftFactor(u_iv_i*), for some u_i,v_i in A*.

Clearly, MFNs generalize PNs. MFNs can also easily simulate the systems of CFSM's studied in [Rosier...84] -- and in fact properly include them [Finkel 86]. A series of definitions is now in order.

3.1. Definitions.

Definition 3.1. A language $L \subseteq A^*$ is *strictly monogeneous* if there exist two words $u,v \in A^*$ such that $L \subseteq \mathrm{LeftFactor}(uv^*)$. A language $L \subseteq A^*$ is *monogeneous* if it is equal to a finite union of strictly monogeneous languages, i.e., $L \subseteq \cup_{i=1,\ldots,k} \mathrm{LeftFactor}(u_i v_i^*)$, for some $u_i,v_i \in A^*$. Let (N,M_0) be a FN and f a fifo of N. Then f is :

(1) *structurally monogeneous* [Finkel 82a] if there exists a word $u_f \in A^*$ such that for every transition $t \in T$, $V(t,f) \in u_f^*$.

(2) *strictly monogeneous* if $L_l(N,M_0,f)$ is strictly monogeneous.

(3) *monogeneous* [Finkel 86] if $L_l(N,M_0,f)$ is monogeneous.

A FN (N,M_0) is *monogeneous* (*structurally monogeneous, strictly monogeneous* respectively) iff each fifo is monogeneous (structurally monogeneous, strictly monogeneous respectively). We abbreviate structurally MFNs as S-MFNs. (end of Definition 3.1.)

An example of a strictly monogeneous FN is provided in Figure 3.1. Also the fifo net shown in Figure 2.1. is such that : $L_l(N,M_0,f_1) = \mathrm{LeftFactor}(ab^*)$ and $L_l(N,M_0,f_2) = \mathrm{LeftFactor}(ab^*)$. Furthermore, f_i, $i>2$, is obviously structurally monogeneous. Hence it too is strictly monogeneous.

A natural problem arises : can we decide whether a FN is monogeneous or not ? Unfortunately, in the general case we cannot. As a matter of fact, the monogeneous problem is undecidable in the framework of FNs [Finkel 86]. But in a particularly important case, we can decide the monogeneous problem.

Theorem 3.1. [Finkel 86] Let K be finite automaton or a PN which only fills words into a fifo f, then the monogeneous problem is decidable.

Moreover, we can find in [Finkel 86] many sufficient and necessary conditions for a FN to be monogeneous.

PNs can be simulated by MFNs where the alphabet for each fifo consists of a single letter. In [Starke 83], it is illustrated how labelled PNs can be used to simulate S-MFNs [Finkel 86]. The translation preserves many properties -- and hence offers an alternate proof strategy for many of the results discussed here (with respect to S-MFNs) -- but not all. Certain language properties are not preserved -- for example, LeftFactor ((abba)*) is a no-labelled S-MFN language but not a no labelled PN language [Finkel 83b].

3.2. Main results concerning Monogeneous Fifo Nets.

In [Finkel 86], it is shown that one can effectively construct coverability graphs for MFNs. Suppose (N,M_0) is monogeneous. Then let $K_{(N,M0)}$ be the associated coverability graph. Let $K_{(N,M0)}$ represent an automaton where the initial marking represents the initial state, all markings represent final states, and the transition function is implicitly given by the graph transitions. It is established :

Theorem 3.2. [Finkel 86] $L(N,M_0) = L(N_p,M_{0p}) \cap L(K_{(N,M0)})$.

Sketch of proof. This says that MFNs can essentially be simulated by their associated PNs operating in conjunction with a finite automaton. The PN controls the lengths of words in the fifos but does not control the firing order. The finite automaton sees that the firing rules are satisfied but does not keep track of word lengths. The inclusion from the left to the right is trivial. The other inclusion is proved by induction on the length of a word in $L(N_p,M_{0p}) \cap L(K_{(N,M0)})$. ■

As a result, a nice consequence concerning deterministic PNs was established.

Definition 3.2. [Vidal-Naquet 81] A deterministic Petri net is a labelled PN (N,M_0,h) such that the morphism $h : T \rightarrow X \cup \{\lambda\}$ (X is a finite alphabet) satisfies the following condition :
For every reachable marking M and transitions t,t' in T, if t and t' are fireable from M then t=t' or $h(t)\neq h(t')$.

Theorem 3.3. [Finkel 86] Every MFN can be simulated by a deterministic PN.

Sketch of proof. To construct the equivalent deterministic PN, one uses the language equation of Theorem 3.2. First one translates the finite automaton $K_{(N,M0)}$ into a deterministic PN . Then since the class of deterministic PN languages is closed under intersection, one can easily generate the equivalent deterministic PN. ■

An example of this construction is portrayed in Figures 3.1 through 3.4.

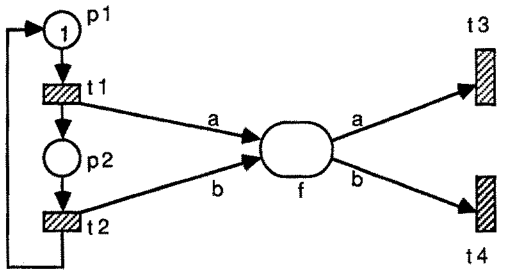

Figure 3.1.

Consider the MFN (N,M_0) shown in Figure 3.1. (N,M_0) is monogeneous because $L_l(N,f)$ is included in LeftFactor((ab)*). The coverability graph $K_{(N,M0)}$ for (N,M_0) is the shown in Figure 3.2.

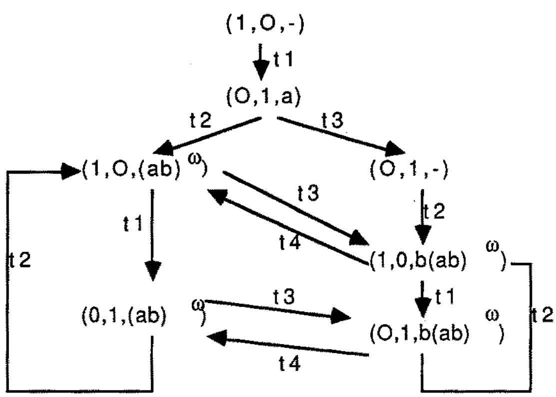

Figure 3.2.

The labelled PN that is equivalent to $K_{(N,M0)}$ is shown in Figure 3.3.

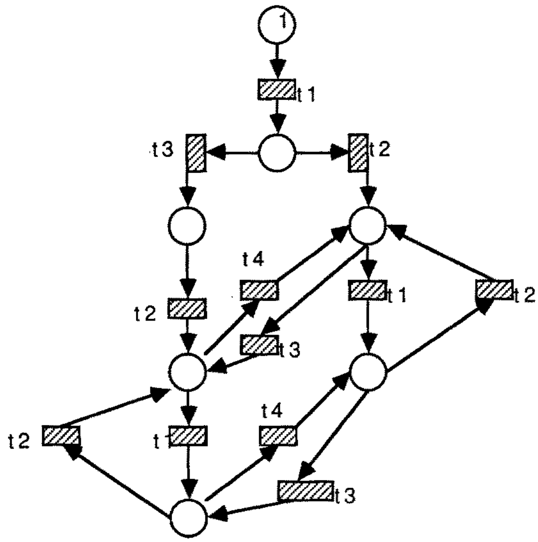

Figure 3.3.

To construct the equivalent deterministic PN, one must generate the intersection of the associated PN (which has the same topology as the MFN shown in Figure 3.1.) with the PN shown in Figure 3.3. In this particular example, it suffices to remark that there is only one constraint to add onto the net shown in Figure 3.3 to obtain the desired intersection :

$$\#(u,t_1) + \#(u,t_2) \geq \#(u,t_3) + \#(u,t_4)$$

where $\#(u,t_1)$ represents the number of t_1s in the word u. We also add a new place p such that : $V(p,t_3) = V(p,t_4) = 1$ and $V(t_1,p) = V(t_2,p) = 1$. As a result we obtain the deterministic PN shown in Figure 3.4.

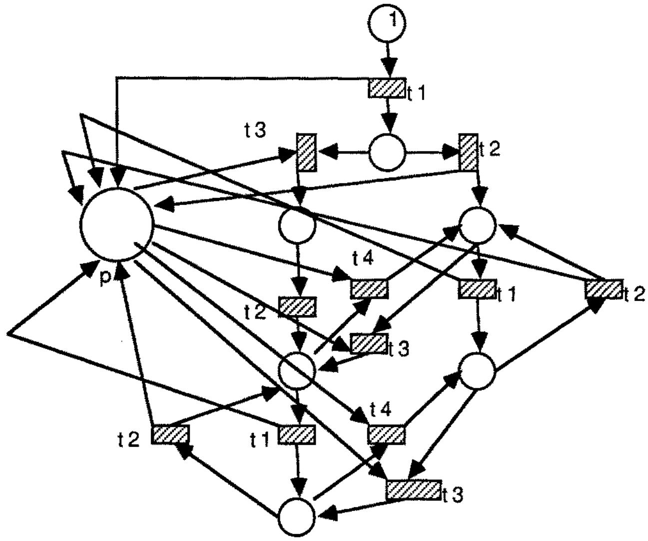

Figure 3.4.

Remark : Another way to associate a labelled PN to a MFN has been given by Starke in the particular case of Structurally MFNs [Starke 83]. His construction likely could be generalized to MFNs.

There are two major interests of the effective construction described in Figure 3.4. First, it shows a deep relation between monogeneous FNs and deterministic PNs : a consequence of Theorem 3.3. is that the class of labelled MFN languages is equal to the class of labelled PN languages. Secondly, it allows to decide the RP, the LP and the RegP by using the fact these properties are preserved by the construction and are decidable for deterministic PNs.

With respect to decidability issues MFNs represent a well behaved class. We obtain :

Theorem 3.4. [Finkel 86] The TDP, the PDP, the BP, the RP, the QLP, the LP, and the RegP are all decidable with respect to MFNs. Furthermore, the center of a MFN is effectively realizable.

Sketch of proof. The coverability graph construction of [Finkel 86] yields straight-forward algorithms for the TDP, the PDP, the BP, and the QLP. The RP and LP (the RegP) can be solved by first constructing the deterministic PN in the fashion just described and then applying the algorithm of Mayr [Mayr 84] (Valk and Vidal-Naquet [Valk ... 81]). A recursive representation for the center can also be derived by applying the strategy provided in Valk and Jensen [Valk ... 85] to the deterministic PN. ∎

These results can be used to upgrade certain results concerning systems of CFSM's. For example, [Rosier ... 84] considers systems of CFSM's where one of the two machines sends only a single type of message. Such systems can be modelled via MFNs since the channel in question is monogeneous. In fact, by allowing the channel to behave in a monogeneous fashion the theorems of [Rosier ... 84] can be strengthened considerably. For example, we can prove the following :

Theorem 3.5. [Finkel 86] The BP is decidable for a system of two CFSM's in which at least one of the fifo channel is monogeneous.

4. Linear Fifo Nets.

In this section, we discuss Linear Fifo Nets (LFNs) and some of their properties. In a LFN the input language of each fifo is contained within some particular linear language (i.e., each input language is included in $a_1^*...a_n^*$, for some $a_1,...,a_n$ in A with $a_i \neq a_j$ for all $i \neq j$). From a theoretical point of view LFNs are a natural generalization of MFNs. A restricted version of LFNs was studied in [Gouda ... 85]. This study concerned systems of CFSM's where each channel was over a bounded language, and provided algorithms to decide the BP and the PDP. LFNs were then introduced in [Choquet ... 87]. This summary closely follows the presentation given there, but considers a slightly more useful version which we call LFNs with a Structured Set of Terminal Markings (SSTM - LFNs). The ideas involved are the same as those employed in [Choquet ... 87]. We shall present the decidability results in the framework of SSTM - LFNs, which are more general than LFNs without terminal markings. As a matter of fact, every LFN without terminal marking can be seen as a particular SSTM - LFN where the set of terminal markings is the reachability set. The reachability set is always a SSTM.

4.1. Definitions.

Definition 4.1. Let A be a finite alphabet. A language L in A^* is said to be *bounded* or *linear* iff L is included in $a_1^*...a_n^*$, for some $a_1,...,a_n$ in A with $a_i \neq a_j$ for all $i \neq j$. A fifo is said to be *linear* iff its input language is bounded. A fifo net is said to be *linear* iff each fifo in the net is linear and has as its initial marking an element of a_1^*.

Let us remark that there is no special constraints on V and Γ: as for MFNs, the unique constraint is on the input language. Of course there exist some structural (on V and Γ) conditions which insure the input language to be linear but it does not exist a sufficient and necessary structural condition for the linear property.

Let us consider the following example [Choquet ... 87] in which the class of linear FNs is used to model and to verify the partial correctness of a parallel program. Observe the sequence of code shown in Figure 4.1. Suppose that we have two sequential processors : one that performs arithmetic operations and one that performs I/O operations. The schemata for an equivalent parallel program might take the form of the LFN shown in Figure 4.2. Actually, the LFN as shown in Figure 4.2. is not yet equivalent--as it permits deadlocks : many deadlocks can occur, for example, any computation in which the transition t_9 occurs before the last occurrence of the transition t_2. We will rectify this later.

```
(1)   n = 1;
(2)   repeat      read (A[n]) ; n := n + 1    until  condition;
(3)   for i := 1 to n do X[i] := 2*A[i];
(4)   for i := 1 to n do  write (X[i]);  read (B[i]);
(5)   for i := 1 to n do  Y[i] := X[i] + B[i];
(6)   for i := 1 to n do  write (Y[i]);
```

Figure 4.1.

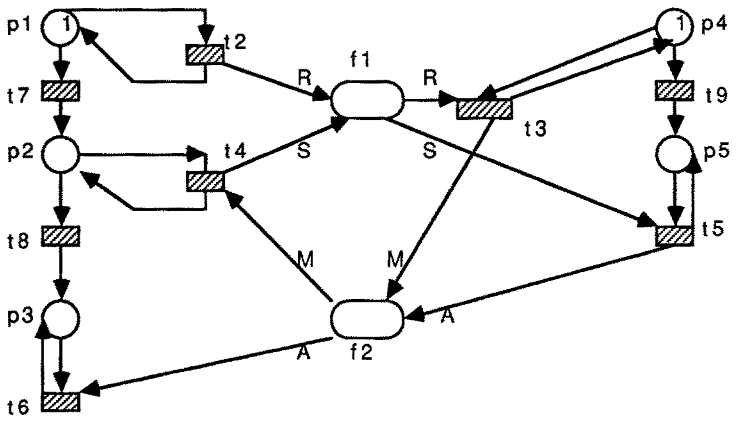

Figure 4.2.

The firing of transitions t_2 (t_3,t_4,t_5,t_6, respectively) corresponds to a single iteration of the loop in statement (2) ((3),(4),(5),(6), respectively). The total number of times t_2 fires corresponds to or determines the value of n. Note that this value is not predetermined. The total number of times t_3 (t_4,t_5,t_6, respectively) fires is then limited by the LFN to n. Since there is a single processor for I/O operations (arithmetic operations), the operations for statements (2),(4), and (6) ((3) and (5)) are performed in order. (Transitions t_7, t_8, and t_9 insure this behavior in the LFN.) The parallelism comes between the two sequential processors. For example, the operations for i:=1 to j for statement (3) ((4),(5),(6), respectively) can overlap with the operations for statement (2) ((3),(4),(5), respectively) for i := j+1 to n for every j, $1 \le j < n$. The LFN will allow precisely this amount of parallelism thus insuring that all data dependencies are observed. The configuration of the fifos, f_1 and f_2, provide information regarding the evolution of the computation. For example, if f_1 contains the word $R^{p1}.S^{p2}$ (i.e., p_1 R's followed by p_2 S's) f_2 contains the word A^{p3} (i.e., p_3 A's) and p_2 and p_4 both contain tokens then we can ascertain that $n = p_1 + p_2 + p_3$ and that the computation has progressed to the point where statement (3) has processed all but the last p_1 iterations and statement (4) has processed exactly p_2 iterations. The fact that $n = p_1 + p_2 + p_3$ serves to illustrate that the parallelism has observed the required data dependencies.

The last point we would like to make is that any computation where t_8 (t_9) fires before t_4 (t_3) has fired exactly n times, does not correspond to a parallel simulation of the program segment in question. In what follows we describe how the LFN model can be constrained in a manner that will force the LFN in Figure 4.2 to only allow computations of the desired type. The idea is straight out of [Choquet ... 87] where it was applied to PNs.

Other linear FNs can be seen, for example, in communication protocols (see [Finkel 88]).

Definition 4.2. Let $H = (N, M_0)$ be a LFN where $N = (F,T,A,V)$. Let $\mathcal{M}$ be a set of markings over F. $\mathcal{M}$ is said to be a *Structured Set of Terminal Markings* (SSTM) with respect to H iff :

(1) membership in $\mathcal{M}$ is decidable,

(2) $M_0 \in \mathcal{M}$,

(3) $\forall x,y \in T^*$ we have that $M_0(xy{>}M$, $M_0(x{>}M'$, and $M \in \mathcal{M}$ implies that $M' \in \mathcal{M}$ (i.e., each marking reached on a path into $\mathcal{M}$ must be in $\mathcal{M}$), and

(4) $\forall x \in T^*$ we have that $M \in \mathcal{M}$, $M(x^i{>}M_i$, $i \ge 1$, $M \le M_1$, and $M_1 \in \mathcal{M}$ implies that $\forall i \ge 1$, $M_i \in \mathcal{M}$. (i.e., any sequence of transitions which when applied to a marking in $\mathcal{M}$ terminates at another marking in $\mathcal{M}$ and can be repeated indefinitely without leaving $\mathcal{M}$).

Definition 4.3. $H_{\mathcal{M}} = (N, M_0, \mathcal{M})$ is now said to be a *LFN having a Structured Set of Terminal Markings* (SSTM-LFN) ; if $H_{\mathcal{M}}$ contains only fifos that are over single letter alphabets (and hence is simply a PN) then $H_{\mathcal{M}}$ is said to be a PN having a Structured Set of Terminal Markings (SSTM-PN). SSTM-PNs were considered in [Choquet ... 87]. The *language* of $H_{\mathcal{M}}$ is $L(N,M_0,\mathcal{M}) = \{x \mid x \in T^*, M_0(x{>}M, M \in \mathcal{M}\}$.

Note that $L(N,M_0,\mathcal{M}) = \mathrm{LeftFactor}(L(N,M_0,\mathcal{M}))$. Because of properties (2) and (3) the *reachability tree* for $H_{\mathcal{M}}$ is simply the reachability tree for H pruned by truncating a path whenever it leaves $\mathcal{M}$. Hence, the *reachability set* of $H_{\mathcal{M}}$, $R(N,M_0;\mathcal{M})$ is equal to $R(N,M_0) \cap \mathcal{M}$. One can read a general discussion about reachability trees in [Finkel 87].

Now let us once again consider the LFN portrayed in Figure 4.2. Let $\mathcal{M}$ be the set of markings where:

$p_4 = 0$ implies $p_1 = 0$ and f_1 contains no R's, and

$p_2 = 0$ implies $p_4 = 0$ and f_2 contains no M's.

Then the resulting SSTM-LFN exactly captures the desired schemata.

4.2. The decidable properties of SSTM-LFNs.

Decision problems with respect to LFNs were considered in [Choquet...87]. There the TDP, the BP, and the RP were all shown to be decidable. In particular, [Choquet ... 87] shows that the language of each LFN is equal to the language of some labelled SSTM-PN, and given a LFN illustrates how to effectively construct the corresponding SSTM-PN.

Theorem 4.1. [Choquet ... 87] The language of each LFN is equal to the language of some labelled SSTM-PN.

Sketch of proof. The general idea is for the corresponding SSTM-PN to simulate a fifo over $a_1^*...a_k^*$ by $2*k$ places ; k places are used for synchronization and k places are used to keep track of the number of a_is, $1 \leq i \leq k$, in the current marking. A structured set of terminal markings is then constructed that forces the SSTM-PN to "faithfully" simulate the LFN.

This construction is such that the TDP, the PDP, the BP, and the RP with respect to LFNs can easily be decided given an algorithm for the respective problems with respect to SSTM-PNs.

It is easy to show the following fact concerning LFNs :

Fact 4.2: If $\mathcal{M}$ and $\mathcal{N}$ are both structured sets of terminal markings with respect to H then so are $\mathcal{M} \cap \mathcal{N}$ and $\mathcal{M} \cup \mathcal{N}$.

As a result, the translation from LFNs to SSTM-PNs employed in [Choquet ... 87] can be generalized for SSTM-LFNs. The constraints which force the SSTM-PN to faithfully simulate the fifos are simply added to the original constraints (now interpreted on the PN, of course). Fact 4.1 insures us that the resulting structure is a SSTM-PN. Hence, a straightforward generalization of the result in [Choquet ... 87] is :

Theorem 4.3. [Choquet...87] The TDP, the PDP, the BP, and the RP are decidable for LFNs providing they are decidable for SSTM-PNs.

From properties (2)-(4) concerning structured sets of terminal markings with respect to SSTM-PNs, [Choquet ... 87] proves the following two facts :

Fact 4.4 : $L(N,M_0,\mathcal{M})$ is infinite iff there exist x in T*, M,M' in $R(N,M_0,\mathcal{M})$ such that M(x>M' and M≤M'.

Fact 4.5 : $R(N,M_0,\mathcal{M})$ is infinite iff there exist x in T* , M,M' in $R(N,M_0,\mathcal{M})$ such that M(x>M' and M<M'.

Using facts 4.4 and 4.5, a simple coverability graph can be constructed for $H_{\mathcal{M}}$. The coverability graph is constructed in the usual fashion. $H_{\mathcal{M}}$ will be unbounded iff an "ω" appears somewhere in the graph, and result in total deadlock iff it is bounded and the graph contains no cycle. Reachability in $H_{\mathcal{M}}$ can be decided using Mayr's algorithm [Mayr 84] in conjunction with an algorithm for membership in T. The PDP can be decided using an algorithm for the reachability of partially described markings. The latter algorithm can be shown to exist (with respect to SSTM-PNs) via the standard reduction to reachability. Hence, we obtain :

Theorem 4.6. [Choquet...87] The TDP, the PDP, the BP, and the RP are decidable for SSTM-PNs.

Corollary 4.7. The TDP, the PDP, the BP, and the RP are decidable for SSTM-LFNs.

Unfortunately, the coverability graph for SSTM-PNs does not possess some properties one might expect in a coverability graph. For example, unless the SSTM-PN under consideration is bounded, a valid firing sequence may not correspond to a path in the graph. See [Choquet ... 87]. Hence, it does not seem to be the case that the same strategy can be employed to solve the QLP. Never the less, the QLP with respect to SSTM-LFNs is decidable. The result follows from the result in [Choquet 87] which shows that the QLP for LFNs is reducible to the RP for LFNs. Hence, we have :

Corollary 4.8. [Choquet 87] The QLP is decidable for SSTM-LFNs.

One might also surmise that the LP is decidable with respect to SSTM-LFNs (because of its equivalence to the RP with respect to general PNs [Hack 76]), but this too does not readily follow from the work in [Choquet ... 87].

5. Topologically Free Choice Fifo Nets.

In this section, we discuss FNs without order deadlocks (i.e., deadlocks caused by the order of the messages in the fifos) and some of their properties. Recall that the defining constraints for MFNs and LFNs were directed at the input language of the fifos. Classes of Topologically Free Choice FNs (TFCFNs) are defined by placing constraints on the output languages of the fifos. We consider three classes of TFCFNs -- Extended-TFCFNs (E-TFCFNs), TFCFNs, and Strict-TFCFNs (S-TFCFNs). E-TFCFNs and TFCFNs were introduced in [Finkel 85] and [Finkel 86] (under the same name minus the word Topologically) ; while S-TFCFNs were first considered in [Choquet 87]. An accounting of the essential properties for each of these three classes can be found in [Finkel ... 87]. One important property is that the PN (respectively free choice PN) languages are a subset of the E-TFCFN (respectively V-TFCFN) languages [Finkel 86] [Finkel...87]. Both inclusions are proper because the anti-Dyck language defined in [Vauquelin ... 80] is a TFCFN (E-TFCFN) language but not a PN language [Finkel 86].

For E-TFCFNs, the LP, the QLP, the TDP, and the CP are known to be decidable [Finkel 86], [Choquet 87]. These results were obtained by considering the relationship between E-TFCFNs and their associated coloured PNs. In this section, we outline this relationship. We also discuss a generalization of Commoner's theorem, for TFCFNs, which can be used to generate a simpler algorithm for the LP [Finkel 86]. For S-TFCFNs, we illustrate another important technique involving residue sets [Valk ... 85]. This strategy can be used to generate a simpler algorithm for the LP, the QLP, and the TDP with respect to S-TFCFNs [Choquet 87] [Finkel...87].

5.1. An Example.

Three processes -- see Figure 5.1 -- Q_1, Q_2, and Q_3 send messages a,b,c through a fifo f to four other processes P_1, P_2, P_3, and P_4. The seven processes are considered like transitions.

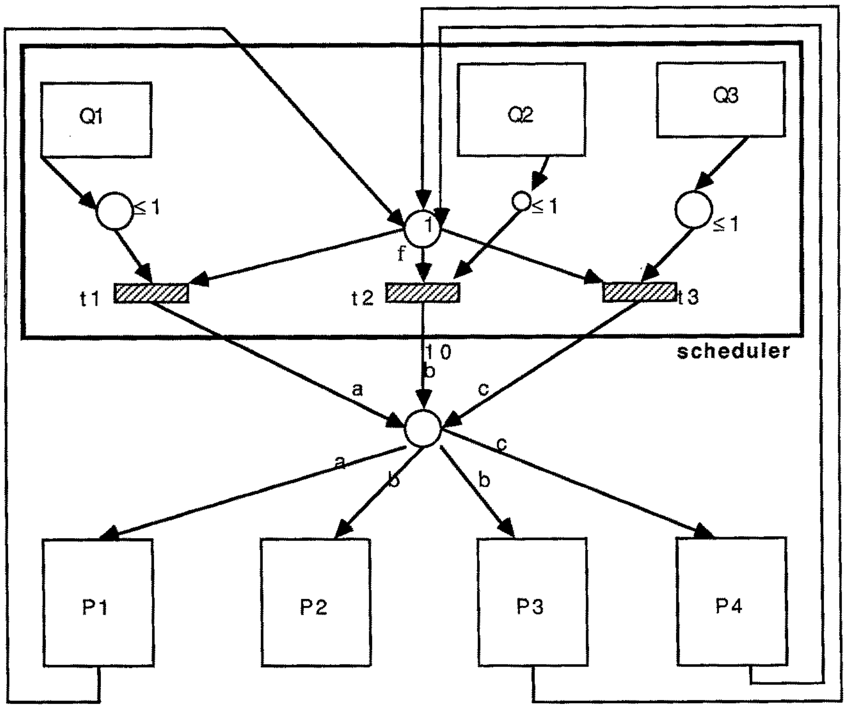

Figure 5.1.

Let us note that b^{10} in Figure 5.1 means b.b...b ,10 times. We assume non-determinism with respect to the sending of messages a,b,c and with respect to processes P_2 and P_3. In this example, the seven processes are considered like transitions. Moreover, Q_1 must run before P_1, Q_2 before P_2 or P_3 , and Q_3 before P_4. Clearly, then one can see that this system is quasi-live and without total deadlock. The system does, however, contain partial deadlocks as $Q_2.t_2.(P_2)^{10}$ is a deadlock (here also $(P_2)^{10}$ represents $P_2.P_2...P_2$, 10 times). This system will turn out to be a TFCFN, hence some of the eight problems can be solved with tools we are going to provide in this section.

5.2. Definitions.

A number of definitions are now in order.

Definition 5.1. [Finkel...87] A FN (N,M_0) is said to be *normal* if the following four conditions are satisfied :

(1) $|A| \leq 2$,

(2) the input and the output alphabets are identical for each fifo f in F,

(3) for all t in $\Gamma(f)$, $V(f,t)$ belongs to A , and

(4) for all f in F, $M_0(f)$ belongs to A_f.

As an example the reader should once again consult the FN portrayed in Figure 2.1 as it is normal.

Now let us recall a definition from [Hack 72] concerning free choice PNs.

Definition 5.2. [Hack 72] A place p in a PN is **free choice** (or satisfies the Hack condition) iff: $|\Gamma(p)| > 1$ implies that for all $t \in \Gamma(p)$, $\Gamma^-(t) = \{p\}$.

Definition 5.3. [Finkel 85] A **TFCFN** (N,M_0) is a FN that satisfies the following three properties :

(1) (N,M_0) is normal,

(2) for all $f \in F$, $|A_f| > 1$ implies that f satisfies the Hack condition, and

(3) the associated PN is a free choice PN.

To extend the definition 5.3., we relax the third condition.

Definition 5.4. [Finkel 86] An **Extended-TFCFN** (N,M_0) is a FN that satisfies the following two properties :

(1) (N,M_0) is normal and,

(2) for all $f \in F$, $|A_f| > 1$ implies that f satisfies the Hack condition.

Definition 5.5. [Choquet 87] A **S-TFCFN** is a TFCFN where every transition in the associated PN has at most one input fifo (i.e., for all $t \in T$, $|\Gamma^-(t)| \leq 1$).

5.3. Main results for E-TFCFNs.

Here we sketch the main known decidable properties for E-TFCFNs. For detailed accounts the reader should consult [Finkel 86], [Choquet 87], and [Finkel...87]. Let (N,M_0) be an E-TFCFN and (N_c,M_{0c}) be the associated coloured PN. An important relationship -- called weak confluence --

between (N,M_0) and (N_c,M_{0c}) was established in [Finkel... 87]. The property is expressed as the following fact.

Fact 5.1. (weak confluence) Let (N,M_0) be an E-TFCFN. For every $u \in L(N_c,M_{0c})$ there exists a sequence $v \in L(N,M_0)$ and $w \in T^*$ such that $uw \in L(N_c,M_{0c})$ and $v \in Perm(uw)$.

The proof of fact 5.1. is rather technical and it can be found in [Finkel...87] or in [Choquet 87]. An important consequence is the following result about the relationship between the FN language and the associated coloured PN language :

Theorem 5.2. [Finkel ...87] Let (N,M_0) be an E-TFCFN. We have the following inclusions :
$L(N,M_0) \subseteq L(N_c,M_{0c}) \subseteq$ LeftFactor(Permutation($L(N,M_0)$)).

Sketch of proof. The first inclusion is trivial because there are fewer constraints on the associated coloured Petri net than on the FN. The second inclusion is a consequence of weak confluence (i.e., Fact 5.1). The proof is an induction on the length of a word in $L(N_c,M_{0c})$. ♠

From theorem 5.2., we deduce :

Theorem 5.3. [Finkel...87] Solving the TDP (QLP, LP, respectively) for (N,M_0), where (N,M_0) is an E-TFCFN, is equivalent to solving the TDP (QLP,LP, respectively) for (N_c,M_{0c}).

Corollary 5.4. [Finkel...87] The TDP, the QLP, and the LP are decidable with respect to E-TFCFNs (TFCFNs, S-TFCFNs, respectively).

An effective recursive construction for the center is also possible with respect to E-TFCFNs [Choquet 87], [Finkel... 87]. The construction is obvious -- given the next theorem.

Theorem 5.5. [Finkel... 87] Let (N,M_0) be an E-TFCFN. We have the following equation :
$C(N,M_0) = L(N,M_0) \cap C(N_c,M_{0c})$.

If (N,M_0) is a TFCFN then a form of Commoner's theorem [Commoner 72] can be applied. This yields a much less complex method for solving the LP if (N,M_0) is a TFCFN (or a S-TFCFN) than the strategy suggested by theorem 5.2. Let (N_{cp},M_{0cp}) be the PN obtained by unfolding (N_c,M_{0c}). (See [Brams 83] for a definition of unfolding.) Let us recall that a *"deadlock"* D is a subset of places such that $\Gamma^-(D) \subseteq \Gamma(D)$; a *trap* T is a subset of places such that $\Gamma(T) \subseteq \Gamma^-(T)$. A trap T is *not deficient for a marking M* if $M(p)>0$ for at least one place p of T. One can show :

Theorem 5.6. [Finkel 86] Let (N,M_0) be a TFCFN. (N,M_0) is live iff each "deadlock" in (N_{cp},M_{0cp}) contains no deficient trap at the initial marking.

Sketch of proof. The theorem is proved by illustrating that (N_{cp},M_{0cp}) is a free choice PN with a homogeneous and non-blocking valuation. The conclusion then follows from [Commoner 72]. ♦

5.4. Strict-Topologically Free Choice Fifo Nets.

Let (N,M_0) be a S-TFCFN. Let QL(N) (NTD(N), L(N), respectively) be the set of markings containing M iff (N,M) is quasi-live (has no total deadlock, is live, respectively). Let us remark that if a transition is fireable from a particular marking then it can be fired from a greater (with respect to the subword relation) marking and that the marking reached in the second case is greater. It follows that QL(N), NTD(N), and L(N) are monotonous [Finkel... 87] and according to Higman's theorem their respective sets of minimal markings, called residues, are finite. This provides a very useful tool for analyzing nets. For example (N,M_0) is quasi-live (is not totally deadlocked, is live, respectively) iff M_0 has as a subword a word in Residue(QL(N)) (Residue(NTD(N), Residue(L(N)), respectively). Furthermore these sets can be computed in polynomial time. Hence [Choquet 87], [Finkel ... 87] obtain :

Theorem 5.7. [Choquet 87] The QLP, the TDP, and the LP are solvable in deterministic polynomial time for S-TFCFNs.

6. Conclusion.

In the previous three sections we have discussed the decidability of eight problems with respect to many subclasses of FNs. The results are summarized in Table 6.1., where "D" ("E") represents "Decidable" ("an effective construction exists"). The "?"s indicate open problems. We conjecture that most of them are decidable but we are able to offer little in the way of proof at this time.

FN classes/Problems	TDP	PTP	BP	RP	QLP	LP	CP	RegP
MFNs	D	D	D	D	D	D	E	D
LFNs and SSTM-LFNs	D	D	D	D	D	?	?	?
E-TFCFNs	D	?	?	?	D	D	E	?

Table 6.1.

This table does not contain Structurally MFNs, TFCFNs and Strict TFCFNs because the main decidability results (not the complexity results !) for these three subclasses are the same than for the more general classes, MFNs and E-TFCFNs.

Acknowledgment : We thank the four anonymous referees for numerous suggestions that led to an improved presentation of our paper.

7. Bibliography.

- [Aho...79] A.V. Aho, J.D. Ullman and M. Yannakakis "Modelling communications protocols by automata" in 20th Annual Symp. on Found. of Comp. Sci. (1979).

- [Berthelot ... 82] G. Berthelot and R. Terrat "Petri nets theory for correctness of protocols" IEEE Trans. on Comm. COM 30 (12), (December 1982).

- [Best... 86] E. Best and C. Fernandez "Notations and terminology on Petri nets" Arbeitspapiere der GMD 195, (January 1986).

- [Boasson...81] L. Boasson and M. Nivat "Centers of languages" L.N.C.S. No. 104, Springer-Verlag, (1981).

- [Bochmann 78] G. Bochmann "Finite state description of communication protocols" Proc. Comp. Network Protocols Symp., pp. 361-371, Liege, Belgium, (February 1978).

- [Bochmann 87] G. Bochmann "Principles of protocols conversion and communication gateways" Report of the Univ. of Montreal, No. 624, (May 1987).

- [Brams 83] G. Brams "Réseaux de Petri: theorie et pratique" T.1. Edition Masson, (1983).

- [Brand...83] D. Brand and P. Zafiropulo "On communicating finite-state machines" J.A.C.M., Vol. 30, No. 2 , pp. 323-342 (April 1983).

- [Brauer ...86a] W. Brauer, W. Reisig and G. Rozenberg (Eds) "Petri Nets : Central models and their properties" Advances in Petri Nets 1986, Part. 1, Proceedings of an Advanced course, LNCS No. 254, Bad Honnef, (September 1986).

- [Brauer ...86b] W. Brauer, W. Reisig and G. Rozenberg (Eds) "Applications and relationships to other models of concurrency" Advances in Petri Nets 1986, Part. 2, Proceedings of an Advanced course, LNCS No. 255, Bad Honnef, (September 1986).

- [Choquet...86] A. Choquet and A. Finkel "Applications of residues for the analysis of parallel systems communicating by Fifo channels" Report LRI No. 278, Univ. Paris 11, Orsay, (May 1986).

- [Choquet...87] A. Choquet and A. Finkel "Simulation of linear fifo nets by Petri nets having a structured set of terminal markings" Report LRI No 324 (January 1987). Also in 8th European Workshop on Applications and theory of Petri nets, Zaragoza, Spain, (June 1987).

- [Choquet 87] A. Choquet "Analyse et propriétés des processus communiquant par files fifo : réseaux à files à choix libre topologique et réseaux à files linéaires" Thèse de 3-ième cycle , Université Paris 11, (25th September 1987).

- [Commoner 72] F. Commoner "Deadlocks in Petri nets" Applied Data research Inc. Wakefiels Mass. CA 7206-2311 (1972).

- [Finkel 82a] A. Finkel "About monogeneous fifo Petri nets" 3rd European Workshop on Applications and theory of Petri nets. Varenna, Italy, (27th-30th September 1982). Also in Report LITP No. 82-40 (1982).

- [Finkel 82b] A. Finkel "Deux classes de réseaux à files: les réseaux monogènes et les réseaux préfixes" Thèse de 3-ème cycle, University Paris 7, Report LITP No. 83-3 (October 1982).

- [Finkel...83a] A. Finkel and G. Memmi "Fifo nets : a new model of parallel computation" Proc. of the 6th G.I. Conf. on Theor. Comp. Sci., RFA, Dortmund, L.N.C.S No. 145, Springer-Verlag, (January 1983).

- [Finkel 83b] A. Finkel "Control of a Petri net by a finite automaton" 3rd Conf. of Soft. Techn. and Theor. Sci., India, Bangalore, (12th-14th December 1983).

- [Finkel 85] A. Finkel "Boundedness and liveness for monogeneous Fifo nets and for free choice Fifo nets - Applications to analysis of protocols" Report LRI, No. 205, (January 1985).

- [Finkel 86] A. Finkel "Structuration des systèmes de transitions : applications au contrôle du parallélisme par files fifo" Thèse d'Etat, University Paris 11, (17th June 1986).

- [Finkel 87] A. Finkel "A generalization of the procedure of Karp and Miller to well structured transition systems" 14th ICALP, Karlsruhe, RFA, (July 1987).

- [Finkel ... 87] A. Finkel and A. Choquet "Fifo nets without order deadlock" Report LRI, No. 356, (1987) and in Acta Informatica 25, pp. 15-36, (January 1988).

- [Finkel 88] A. Finkel "Specification and validation of communication protocols using fifo nets". Submitted (January 1988).

- [Flé...85] M. Flé and G. Roucairol "Fair serializability of iterated transactions using FIFO-nets" Advances in Petri nets L.N.C.S. No. 188, Springer-Verlag, (1985).

- [Gouda...85] M. Gouda, E. Gurari, T. Lai and L. Rosier "On deadlock detection in systems of communicating finite state machines" Computers and Artificial Intelligence, Vol. 6, No. 3, pp. 209-228. (1987).

- [Hack 72] M. Hack "Analysis of production schemata by Petri nets" M.S. Thesis - Dept. Elect. Eng. M.I.T. Camb. Mass. Proj. MAC.MAC-TR 94 (September 1972).

- [Hack 76] M. Hack "Decidability questions for Petri nets", MIT, LCS, TR 161, Cambridge, MA, (1976).

- [Higman 52] G. Higman "Ordering by divisibility in abstract algebras" Proc. London Math. Soc. 2 (1952).

- [Jantzen 79] M. Jantzen "On the hierarchy of Petri net languages" R.A.I.R.O. Vol. 13 No.1 pp. 19-30 (1979).

- [Jensen 86] K. Jensen "High level Petri nets" Advances in Petri Nets 1986, Part. 2, Proceedings of an Advanced course, LNCS No. 255, Bad Honnef, (September 1986).

- [Karp...69] R. Karp and R. Miller "Parallel program schemata" J.C.S.S.: 3 pp. 147-195 (1969).

- [Kosaraju 82] S. Kosaraju "Decidability of reachability in vector addition systems" in Proc. 14th Ann. ACM Symp. on Theor. of Comp. (1982).

- [Martin...81] R. Martin and G. Memmi "Specification and validation of sequential processes communicating by fifo channels" 4th Int. Conf. on Soft. Eng. for Telecom. Switching Syst. IEE Warwick pp. 54-57, (July 1981).

- [Mayr 84] E. Mayr "An algorithm for the general Petri net reachability problem" in SIAM J. Comp. Vol. 13 (3) (August 1984).

- [Memmi 83] G. Memmi "Méthodes d'analyse de réseaux de Petri, réseaux à files, et applications aux systèmes temps-réel" Thèse d'Etat, University Paris 6, (June 1983).

- [Memmi...85] G. Memmi and A. Finkel "An introduction to Fifo nets - monogeneous nets : a subclass of Fifo nets" T.C.S. 35 (1985).

- [Memmi...86] G. Memmi and J. Vautherin "Analysing nets by the invariant method" in "Applications and relationships to other models of concurrency" Advances in Petri Nets 1986, Part. 2, Proceedings of an Advanced course, LNCS 255, Brauer(Eds), Bad Honnef, (September 1986).

- [Peterson 81] J. Peterson "Petri net theory and the modelling of systems" Prentice-Hall, (1981).

- [Peterson... 85] J. Peterson and A. Silberschatz "Operating system concepts" Addison-Wesley (1985).

- [Petri 73] C. A. Petri "Concepts on Net Theory" M. F. C. S. Proceedings of Symp. and Summer school, High Tatras pp. 137-148 (1973).

- [Petri 86] C. A. Petri "Concurrency theory" in "Petri Nets : Central models and their properties" Advances in Petri Nets 1986, Part. 1, Proceedings of an Advanced course, LNCS No. 254, Bad Honnef, (September 1986).

- [Reisig 85] W. Reisig "Petri nets", Springer-Verlag, (1985).

- [Rosier...84] L. Rosier and M. Gouda "On deciding progress for a class of communicating protocols" Proceed. of the 8th Ann. Conf. on Inf. Sci. and Syst. Princeton Univ. pp. 663-667 (1984).

- [Rosier...86] L. Rosier and H. Yen " Boundedness, empty channel detection, and synchronisation for communicating finite automata" T.C.S. 44 pp. 69-105 (1986).

- [Roucairol 86] G. Roucairol "FIFO Nets" in "Petri Nets : Central models and their properties" Advances in Petri Nets 1986, Part. 1, Proceedings of an Advanced course, LNCS No. 254, Bad Honnef, (September 1986).

- [Sarikaya...86] B. Sarikaya and G. Bochmann (Editors) "Protocol Specification, Testing, and Verification, 6" , IFIP 86, Noth Holland (1987).

- [Starke 83] P. Starke "Monogeneous Fifo nets and Petri nets are equivalent" Bulletin of EATCS 21 (1983).

- [Valk...85] R. Valk and M. Jantzen "The residue of vector sets with applications to decidability problems in Petri nets" Acta Informatica 21 (1985).

- [Valk...81] R. Valk and G. Vidal-Naquet "Petri nets and regular languages" J.C.S.S. 23 (3) pp. 229-325 (1981).

- [Vauquelin...80] B. Vauquelin and P. Franchi-Zannettacci "Automates à files" T.C.S. 11 (1980).

- [Vidal-Naquet 81] G. Vidal-Naquet "Rationalité et déterminisme dans les réseaux de Petri" Thèse d'Etat University Paris 6, (April 1981).

- [Zafiropulo...86] P. Zafiropulo and AL "Towards analyzing and synthesizing protocols" IEEE Trans. on Comm., Vol. COM-28, No. 4, pp. 651-661, (April 1980).

- [Zhao...86] Z. Zhao and G. Bochmann "Reduced reachability analysis of communication protocols : a new appproach" Report of the Univ. of Montréal, No. 570, Canada, (July 1986) also in IFIP 6, North Holland (1987).