TOWARDS A HIERARCHY OF NETS

by

Sarah Porat* and Michael Yoeliφ

Technical Report #224
November 1981

* This research is part of the M.Sc. thesis of the first author

φ The research of the second author was supported by the Fund for the Promotion of Research at the Technion, Grant 120-589.
ABSTRACT

This paper deals with various extensions of the concept of Petri net. It uses net languages in order to compare the capabilities and limitations of various net models. The paper extends earlier work by Agerwala and Flynn.
1. INTRODUCTION

In [AG-FL] several extensions of the concept of Petri net are introduced. The authors define a language ("simulation set") for every net and use these languages to establish a hierarchy of "capabilities" among the net models introduced. Their work relies to some extent on results obtained in [KOS].

In this paper we solve an open problem of [AG-FL], namely the relationship between Petri-Nets and Log-Nets. We also formalize and simplify proofs appearing in [AG-FL] and [KOS]. Moreover, we formulate and solve additional problems related to the hierarchy of various net models.
2. INHIBITOR NETS - BASIC DEFINITIONS

Definition 2.1: An I-Net (Inhibitor Net) is a 4-tuple \( N = (P, T, V, I) \), where

1. \( P \) and \( T \) are finite sets of places and transitions, respectively.
2. \( P \cap T = \emptyset, T \neq \emptyset \).
3. \( V \) is a function,
   \[ V: (P \times T) \cup (T \times P) \rightarrow \{0, 1\} \]
4. \( I \) is a function,
   \[ I: P \times T \rightarrow \{0, 1\} \]

The function \( I \) satisfies the condition

\[ (\forall p \in P)(\forall t \in T) \ I(p, t) \times V(p, t) = 0 \]

Definition 2.2: A P-Net (Petri Net) is an I-Net, satisfying the condition \( I(P \times T) = \{0\} \).

For P-Nets we simply write \( N = (P, T, V) \).

We denote by \( \omega \) the set of non-negative integers.

Definition 2.3: A marked I-Net is a pair \( S = (N, M) \), where \( N \) is an I-Net and \( M \) is a marking of \( N \), i.e. a function \( M: P \rightarrow \omega \).

A marked I-Net \( S = (P, T, V, I, M) \) is represented graphically as follows:

1. Places are represented by circles (\( \circ \))
2. Transitions are represented by bars (\( \rightarrow \))
3. The function \( V \) is represented by directed arcs in the usual way.
4. The function \( I \) is represented by arcs shown thus: \( \circ \).
5. The integer \( m = M(p) \) is represented by \( m \) dots (tokens) inside the place \( p \) or alternatively by writing the integer \( m \) inside the place \( p \). Usually, one omits writing 0 inside a place.
Definition 2.4: Let \( S = (P,T,V,I,M) \) be a marked I-Net. A transition \( t \in T \) is enabled iff the following conditions are satisfied:

\[
\begin{align*}
(1) & \quad (\forall p \in P) \ [M(p) \geq V(p,t)] \\
(2) & \quad (\forall p \in P) \ [I(p,t) = 1 \Rightarrow M(p) = 0].
\end{align*}
\]

Definition 2.5: Let \( S = (P,T,V,I,M) \) be a marked I-Net, and \( t \in T \) an enabled transition of \( S \). We say that the marking \( M' \) of \( N = (P,T,V,I) \) is obtained from \( M \) by firing \( t \) (notation: \( M[t \rightarrow M'] \)), iff \( M' \) satisfies the following condition:

\[
(\forall p \in P) \ M'(p) = M(p) + V(t,p) - V(p,t).
\]
3. INHIBITOR NET LANGUAGES

With a given marked I-Net \( S = (P, T, V, I, M) \) we associate a language \( L(S) \) over the alphabet \( T \) as follows:

**Definition 3.1:** Let \( N = (P, T, V, I) \) be an I-Net. A firing sequence \( x \in T^* \) from the marking \( M \) of \( N \) to the marking \( M' \) of \( N \) (notation: \( M[x > M'] \)) is defined recursively as follows:

1. \( M[\lambda > M] \), where \( \lambda \) denotes the empty sequence.
2. Let \( x = yt \), where \( y \in T^* \) and \( t \in T \). Then \( M[x > M'] \) iff \( M[y > M'] \) and \( M'[t > M'] \) for some marking \( M' \) of \( N \).

**Definition 3.2:** Let \( S = (P, T, V, I, M) \) be a marked I-Net. We define its language \( L(S) \) as follows:

\[
L(S) = \{ x \in T^* \mid (\exists M') M[x > M'] \}.
\]

We now introduce the concept of labelled I-Net (LI-Net), which provides a more general way of associating languages with I-Nets.

**Definition 3.3:** A marked LI-Net is a triple \( \Gamma = (S, \Sigma, \eta) \), where \( S \) is a marked I-Net, \( \Sigma \) is a finite alphabet and \( \eta \) is a mapping \( \eta: T \rightarrow \Sigma \cup \{\lambda\} \).

The language of \( \Gamma \) is defined by:

\[
L(\Gamma) = \eta(L(S)) = \{ \eta(x) \mid x \in L(S) \}.
\]

A marked LP-Net is defined similarly.

By now, we have defined 4 types of nets, namely P-Nets, I-Nets, LP-Nets, and LI-Nets. Let \( X \) stand for P, I, LP, or LI.

A language \( L \subseteq \Sigma^* \) is \( X \)-realizable iff \( L = L(S) \) for some marked \( X \)-Net \( S \). We denote by \( XL \) the set of all \( X \)-realizable languages.
4. HIERARCHY OF I-NETS

Let $X$ and $Y$ be two net types. We write $X + Y$, iff $X_L \not\subseteq Y_L$ and $X \prec Y$, iff $X_L$ and $Y_L$ are not comparable, i.e. we have neither $X_L \subseteq Y_L$ nor $Y_L \subseteq X_L$.

Figure 4.1 indicates the hierarchy among the four net types defined so far. This hierarchy will be proven in this section.

A language $L$ is a prefix language iff

$$(\forall x \in L) \text{pref}(x) \subseteq L$$

where \text{pref}(x) denotes the set of all prefixes of $x$. Clearly, all net languages defined above are prefix languages.

**Definition 4.1:** A prefix language $L \subseteq \Sigma^*$ is prefix-permutable iff the following condition is satisfied.

For every $x \in \Sigma^+$, $\sigma \in \Sigma$, and $y \in \text{perm}(x)$, where $\text{perm}(x)$ denotes the set of all permutations of $x$, we have:

$$x \sigma \in L \land y \in L \Rightarrow y \sigma \in L.$$

**Lemma 4.1:** Every language $L \in IL$ is prefix-permutable.
Proof: Let $S = (P, T, V, I, M)$ be a marked $P$-Net and let $L = L(S)$. We set $M[x \cdot M_x', x \in L$. Assume now $x \sigma \in L$, $y \in \text{perm}(x)$, and $y \in L$. Since $L$ is a prefix language, $x \in L$. One easily verifies that $M_x = M_y$. Since $\sigma$ is enabled for $M_x$, it follows that $y \sigma \in L$. Thus, $L$ is prefix-permutable.

Theorem 4.1: $P \rightarrow LP$

Proof: Let $S = (P, T, V, M)$ be a marked $P$-Net. The marked $LP$-Net $\Gamma = (S, T, \eta)$, where $\eta$ is the identity mapping on $T$, satisfies the condition $L(\Gamma) = L(S)$. Thus $PL \subseteq LPL$. Consider now the marked $LP$-Net $\Gamma$ of Figure 4.2. Since $abc \in L(\Gamma)$, $ba \in L(\Gamma)$, but $bac \in L(\Gamma)$, $L(\Gamma)$ is not prefix-permutable. Hence, by Lemma 4.1, $L(\Gamma)$ $\notin PL$ (obviously $PL \subseteq IL$). Thus, $PL \nsubseteq LPL$, i.e., $P \rightarrow LP$.

Remark: Every finite prefix language is LP-realizable. The LP-Net can be constructed in tree-form, as illustrated by Figure 4.2.

![Figure 4.2](image)

Figure 4.2. Example of a marked LP-Net.
In order to prove $P \rightarrow I$ we need the following two lemmas.

**Definition 4.2:** Let $f_1, f_2$ be two functions, each from the same domain $D$ into $\omega$. We write $f_1 \leq f_2$ iff $\forall d \in D \ f_1(d) \leq f_2(d)$.

**Lemma 4.2:** Let $F$ be an infinite sequence of functions, $F = (f_1, f_2, \ldots)$, each from a finite domain $D$ into $\omega$; then there exist two functions $f_i, f_j \in F$, $i \neq j$, s.t. $f_i \leq f_j$.

For a proof of this lemma we refer to [KOS]. See also [KA-MI].

**Lemma 4.3:** Let $N = (P, T, V)$ be a $P$-Net. Let $M_1, M_2$ be markings of $N$ such that $M_1[y_1] > M_1$ and $M_2[y_2] > M_2$, where $y_1 \in T^*$. Assume $M_1 \leq M_2$.

Then for every $z \in T^+$ if $y_1 z \in L(N, M)$ then $y_2 z \in L(N, M)$.

**Proof:** We use induction on the length of $z$.

**Basis:** $|z| = 1$, i.e. $z \in T$. This case can be immediately verified.

**Induction Step:** Let $z = tz'$, where $z' \in T^*$, $t \in T$. Since $L(N, M)$ is a prefix language and $y_1 z \in L(N, M)$, we have $y_1 t \in L(N, M)$. By induction hypothesis $y_2 t \in L(N, M)$. Let $M_1[t] > M_3$ and $M_2[t] > M_4$. One easily verifies by Def. 2.5 that $M_1 \leq M_2$ implies $M_3 \leq M_4$. By again implying the induction hypothesis, the lemma follows.

The following theorem is a modified version of a result in [KOS].

Our proof is more formal and simpler than the proof in [KOS].

**Theorem 4.2:** $P \rightarrow I$.

**Proof:** By definitions of $P$-Net and $I$-Net, obviously $PL \subseteq IL$.

Consider, now, the marked $I$-Net, $S$, of Fig. 4.3.

For every $i \geq 1$, $a^i \in L(S)$. Let $M[a^i] > M_1$. We imply the Lemma 4.2 to the infinite sequence $(M_1, M_2, \ldots \ldots)$, thus $M_i \leq M_j$ for some $i, j \neq j$. 

Technion - Computer Science Department - Technical Report CS0224 - 1981
Now, \( a^i b^i c \in L(S) \) but \( a^i b^i c \notin L(S) \); hence, by Lemma 4.3
\[ L(S) \notin PL. \]

Thus, \( PL \notin IL \), i.e. \( P+I \).

![Diagram](image)

**Figure 4.3** - Example of a marked I-Net.

Next, we want to prove that \( I \rightarrow LP \). For this purpose we need the following modified version of Lemma 4.3.

**Lemma 4.4:** Let \( \Gamma = (P,T,V,M,E,n) \) be a marked LP-Net. Let \( M_1, M_2 \)
be markings of \( (P,T,V) \) such that \( M[x_1>M_1, M[x_2>M_2, n(x_1) = y_1 \)
and \( n(x_2) = y_2 \). \( x_1 \in T^* \), \( x_1 \in T^* \). Assume \( M_1 \leq M_2 \).

If \( y_1 z \in L(\Gamma) \), and for some marking \( M_3 \) of \( (P,T,V) \) and some
\( w \in T^* \) we have \( M_1[w>M_3, n(w) = z \), then \( y_2 z \in L(\Gamma) \).

**Proof:** Clearly, \( x_1 w \in L(P,T,V,M) \). By Lemma 4.3, \( x_2 w \in L(P,T,V,M) \),
hence \( n(x_2 w) = y_2 z \in L(\Gamma) \).

**Theorem 4.3:** \( LP \rightarrow I \).

**Proof:** (a) We show that the marked I-Net, \( S \), of Fig. 4.3 yields
\( L(S) \notin LPL \).

Indeed, assume there exists a marked LP-Net \( \Gamma = (N',M',T,n) \)
g.t. \( L(\Gamma) = L(S) \). For every \( i \geq 1 \), \( a^i \in L(\Gamma) \), and \( a^i b^i c \in L(\Gamma) \).
Let \( x_i \in L(N', M') \) such that \( M'[x_i \rightarrow M'_i , \ n(x_i) = a^i \) and for some marking \( M'_i, M'_i[w_i \rightarrow M'_i] \) where \( n(w_i) = b^i c \).

We again apply the Lemma 4.2 to the infinite sequence \( (M'_1,M'_2,\ldots) \), thus \( M'_i \leq M'_j \) for some \( i \neq j \). By Lemma 4.4 \( a^i b^i c \in L(\Gamma) \), but \( a^i b^i c \notin L(S) \), contradicting our assumption \( L(\Gamma) = L(S) \). This completes part (a) of the proof, so \( IL \subseteq LPL \).

(b) Consider the marked LP-Net, \( \Gamma \), of Fig. 4.2. \( L(\Gamma) \) is not prefix-permutable (see proof of Theorem 4.1), hence by Lemma 4.1 \( L(\Gamma) \notin IL \). This shows that \( LPL \notin IL \).

\textbf{Corollary 4.1:}

\begin{enumerate}
\item[(a)] \( I + LI \)
\item[(b)] \( LP + LI \).
\end{enumerate}

\textbf{Proof:} Evidently \( LPL \subseteq LIL \) and \( IL \subseteq LIL \). By Theorem 4.3 corollary follows.
S. LOG-NETS - BASIC DEFINITIONS

Definition 5.1: An L-Net (Log-Net) is a 4-tuple \( N = (P,T,V,H) \), where
\((P,T,V)\) is a P-Net,
\(H\) is a function
\(H: P \times T \rightarrow \{0,1\}\).
\(\forall p \in P, \forall t \in T \quad V(p,t) \times H(p,t) = 0\).

A marked L-Net is defined similarly to Definition 2.3.

The graphical representation of an L-Net is that of the P-Net with additional arcs representing \(H\) which are shown thus: 

Definition 5.2: Let \( S = (P,T,V,H,M) \) be a marked L-Net. A transition \( t \in T \) is enabled iff the following conditions are satisfied:
1. \( \forall p \in P \ [M(p) \geq V(p,t)] \)
2. \( [(\exists p \in P) H(p,t) = 1] \Rightarrow [(\exists p \in P) (H(p,t) = 1 \land M(p) > 1)] \).

Definition 5.3: Let \( S = (P,T,V,H,M) \) be a marked L-Net, and \( t \in T \) an enabled transition of \( S \). We say that the marking \( M' \) of \( N = (P,T,V,H) \) is obtained from \( M \) by firing \( t \) (notation: \( M[t \rightarrow M'] \)), iff the following conditions are satisfied:
1. \( (\forall p \in P) (H(p,t) = 0) \Rightarrow M'(p) = M(p) - V(p,t) + V(t,p) \)
2. \( (\forall p \in P) (H(p,t) = 1 \land M(p) = 0) \Rightarrow M'(p) = V(t,p) \)
3. \( (\forall p \in P) (H(p,t) = 1 \land M(p) > 0) \Rightarrow M'(p) = M(p) - H(p,t) + V(t,p) \).

Log-Net Languages

With a given marked L-Net, \( S \), we associate a language \( L(S) \) similarly to Definitions 3.1, 3.2.

We denote by \( LL \) the set of all languages realizable by L-Nets.
6. HIERARCHY OF L-NETS AND I-NETS

Figure 6.1 indicates the hierarchy among the following four net types: P-Nets, I-Nets, LI-Nets and L-Nets.

![Hierarchy of P-, I-, LI-, and L-Nets.](image)

The question whether $P \rightarrow L$ was raised in but left unanswered in [AG-FL]. In this section we prove that indeed $P \rightarrow L$. [AG-FL] also dealt with the comparison between $L$ and $LI$. They indicated informally that every L-Net can be converted into an equivalent LI-Net. However, one can show that their construction is not correct. In this section we provide an improved conversion algorithm, and show that $L \rightarrow LI$. Our result $L \rightarrow I$ is new.

**Theorem 6.1**: $P \rightarrow L$.

**Proof**: Trivially $PL \subseteq LI$.

Now consider the marked LI-Net, $S$, of Figure 6.2. We have that $ab \in L(S)$, $bac \in L(S)$ but $abc \notin L(S)$, hence $L(S)$ is not prefix-permutable, thus by Lemma 4.1 $L(S) \notin PL$.

**Theorem 6.2**: $L \rightarrow I$.

**Proof**: Let $S$ be the marked LI-Net of Fig. 6.2. Since, $L(S)$ is not prefix-permutable, $L(S) \notin IL$, by Lemma 4.1.
One easily verifies that Lemma 4.3 also applies to L-Nets. Hence the argument used in the proof of Theorem 4.2 to show that there exist a marked L-Net, $S'$, such that $L(S') \notin PL$, can be implied to show that $L(S') \notin LL$.

This completes the proof of Theorem 6.2.

Theorem 6.3: \( L \rightarrow LI \).

Proof: Following [AG-FL], we introduce an algorithm which converts every marked L-Net, $S = (P,T,V,H,M)$ into an equivalent marked LI-Net, $\Gamma = (P',T',V',I,M',T,n)$. This algorithm consists of two parts. In its first part, each transition $t \in T$ is transformed locally, namely the "typical" transition of Figure 6.3 is replaced by Fig. 6.4. This is in accordance with [AG-FL]. The second part consists of adding inhibitor arcs from all places in $P' - P$ to all transitions except to their own output transitions.

It is shown in [POR] that the second part of the algorithm is indeed essential. For example, the application of part one only to the marked
L-Net of Figure 6.5 will yield an incorrect result.

Clearly, by Theorem 6.2, $IL \neq LIL$.

**Figure 6.3** - A "typical" transition of a marked L-Net.

**Figure 6.4** - Conversion of Fig. 6.2 into a marked LI-Net.
REFERENCES


