数理解析研究所講究録
第 809 巻 1992 年 130-142

Equivalence between a Finite Automaton and a Petri Net

高橋真吾 (TAKAHASHI Shingo)
Tokyo Institute of Technology

1 Introduction

It is informally noticed that the description ability of the class of Petri nets including even unbounded Petri nets is greater than that of the class of finite automata, and the class of bounded Petri nets has the equal description ability with the class of finite automata.[1,5] These are indeed shown by constructing an "equivalent" Petri net to a given finite automaton. However, the formal meaning of "equivalent" is still left ambiguous. In systems theory "equivalent" models mean, in a formal way, that a homomorphism can be constructed between them, for example between two finite automata. In our case in question, since a Petri net and an automaton are of different "types", no homomorphism can be constructed between them.

The purpose of this paper is to show, by constructing an F-morphism which allow us to consider the equivalence of system models of different types, that (1) a finite automaton can be embedded into a Petri net, and the Petri net preserves all the properties of the embedded automaton; and (2) the behavioral properties of a bounded Petri net are equivalent to those of a finite automaton, especially of a state-transition model.

An F-morphism in this paper is introduced to investigate the similarity between system models of different types. It includes a homomorphism as its special case, and satisfies some similar properties to a homomorphism, for example, F-morphism Theorem which corresponds to Homomorphism Theorem.

2 System Model and F-morphism

In this paper we use a first order logic as a model description language for Petri nets and finite automata. Within that framework we define a system model and an F-morphism between system models.

Definition 2.1 **System Model**

A *system model* M is composed of:
(1) a base set $|M|$;
(2) a set of $\lambda(i)$-ary relations on $|M|$,$\{R_i|i \in I\}$, where $\lambda : I \to N^+$;

(3) a set of $\mu(j)$-ary functions on $|M|$,$\{f_j|j \in J\}$, where $\mu : J \rightarrow N$.

We write M as follows:

$$M =< |M|; \{R_i|i \in I\}, \{f_j|j \in J\} >$$

In this paper we describe the properties of a system model with a first order language **L**, which is of high potential of description and well explored in itself.

The symbols used in **L** are the language of M:

$$\mathbf{L}(M) = \{\{\mathsf{R_i}|i \in I\}, \{\mathsf{f_j}|j \in J\}\}$$

$\{\forall, \wedge, \neg, =\}$, variables, parenthesis and commas. We denote the set of variables by V.

A property of M is expressed by a formula which consists of some terms corresponding to "words". Terms and formulas are recursively defined:

term

(1) variables and constants are terms,

(2) if $t_1, \ldots, t_{\mu(j)}$ are terms, then $\mathsf{f_j}(t_1, \ldots, t_{\mu(j)})$ is term.

formula

(1) if $t_1, \ldots, t_{\lambda(i)}$ are terms, then $\mathsf{R_i}(t_1, \ldots, t_{\lambda(i)})$ and $(t_1 = t_2)$ are formulas,

(2) if ϕ, ψ are formulas, then $(\phi \wedge \psi)$,$\neg\phi$,$\forall \mathsf{x} \phi$ are formulas(where $\mathsf{x} \in V$).

A formula of the form (1) is said to be *atomic formula.* A formula ϕ is said to be a *sentence* if all the variables occuring in ϕ are bounded.

For the following discussion, we define some notations.

For any formula ϕ and any unary relation Q,

$$(\forall \mathsf{x} \in \mathsf{Q})\phi \equiv (\forall \mathsf{x})(\mathsf{Q}(\mathsf{x}) \rightarrow \phi).$$

For an atomic formula $\Phi(\mathsf{x}_1, \ldots, \mathsf{x}_n)$, $\Phi(\mathsf{t}_1, \ldots, \mathsf{t}_n)$ denotes the atomic formula obtained from $\Phi(\mathsf{x}_1, \ldots, \mathsf{x}_n)$ by replacing $\mathsf{x}_1, \ldots, \mathsf{x}_n$ with $\mathsf{t}_1, \ldots, \mathsf{t}_n$.

If an assignment function of the set V of variables to $|M|$ is defined and the value of $|M|$ is assigned by the assignment function to every variable occuring in a formula ϕ, then we can determine whether ϕ holds in $|M|$ or not. If ϕ holds in $|M|$ with an assignment function ρ, we write $M \models \phi[\rho]$. Thus we can consider a property of a model M as a formula ϕ expressing the property which holds in M with some ρ. As for a sentence ϕ, if ϕ holds with some ρ, then it holds with any ρ. The detail should be consulted to the reference.[4] In this paper we represent a property of a model by a sentence.

Definition 2.2 **Basic Morphism**

Let

$$M_1 =< |M_1|; \{R_i^1 \mid i \in I_1\}, \{f_j^1 \mid j \in J_1\} >,$$

$$M_2 =< |M_2|; \{R_i^2 \mid i \in I_2\}, \{f_j^2 \mid j \in J_2\} >$$

be system models. Their types may be different. A function I_O of $|M_1|$ to $|M_2|$ is said to be *basic morphism of* $|M_1|$ *to* $|M_2|$ if the following conditions are satisfied.

There exists a function Bas of $\mathbf{L}(M_1)$ to the set of formulas of $\mathbf{L}(M_2)$ such that for any symbols $\mathsf{R}_i^1, \mathsf{f}_j^1 \in \mathbf{L}(M_1)$:

(1) if $M_1 \models \mathsf{R}_i^1(\mathsf{x}_1, \ldots, \mathsf{x}_{\lambda_1(i)})[\rho]$, then $M_2 \models Bas(\mathsf{R}_i^1)(\mathsf{x}_1, \ldots, \mathsf{x}_{\lambda_1(i)})[I_O \circ \rho]$,

where $Bas(\mathsf{R}_i^1)$ is $(\lambda_1(i))$-ary formula (which means that the free variables occuring in the formula $Bas(\mathsf{R}_i^1)$ are $\mathsf{x}_1, \ldots, \mathsf{x}_{\lambda(i)}$), and $I_O \circ \rho$ denotes the composition of I_O and ρ;

(2) if $M_1 \models (\mathsf{f}_j^1(\mathsf{x}_1, \ldots, \mathsf{x}_{\mu_1(j)}) = \mathsf{x}_{\mu_1(j)+1})[\rho]$, then

$$M_2 \models Bas(\mathsf{f}_j^1)(\mathsf{x}_1, \ldots, \mathsf{x}_{\mu_1(j)+1})[I_O \circ \rho],$$

where $Bas(\mathsf{f}_j^1)$ is $(\mu_1(j)+1)$-ary formula and satisfies the following condition on functions:
$M_2 \models (\forall \mathsf{x}_1 \ldots \mathsf{x}_{\mu_1(j)})(\exists \mathsf{x}_{\mu_1(j)+1})(\forall \mathsf{y}_{\mu_1(j)+1})$

$$(Bas(\mathsf{f}_j^1)(\mathsf{x}_1, \ldots, \mathsf{x}_{\mu_1(j)}, \mathsf{y}_{\mu_1(j)+1}) \leftrightarrow \mathsf{x}_{\mu_1(j)+1} = \mathsf{y}_{\mu_1(j)+1}).$$

$Bas(\mathsf{R}_i^1)$ and $Bas(\mathsf{f}_j^1)$ are called *basic interpretations* of R_i^1 and f_j^1 respectively. The identity $=$ is interpreted as the identity of $\mathbf{L}(M_2)$: $Bas(=_{\mathbf{L}(M_1)}) \equiv =_{\mathbf{L}(M_2)}$.

For any atomic formula $\Phi(\mathsf{t}_1, \ldots, \mathsf{t}_n)$ of a given language, we define the set $T(\Phi)$ of terms as follows.

(1) If Φ is in the form $\mathsf{f}_j(\mathsf{u}_1, \ldots, \mathsf{u}_{\mu(j)}) = \mathsf{x}$ (or equivalently $\mathsf{x} = \mathsf{f}_j(\mathsf{u}_1, \ldots, \mathsf{u}_{\mu(j)})$), where $\mathsf{u}_1, \ldots, \mathsf{u}_{\mu(j)}$ are terms and $\mathsf{x} \in V$,

$$T(\Phi) = \{\mathsf{u}_k \mid \mathsf{u}_k \notin V, k = 1, \ldots, \mu(j)\};$$

(2) otherwise,

$$T(\Phi) = \{\mathsf{t}_k \mid \mathsf{t}_k \notin V, k = 1, \ldots, n\}.$$

In the form (1) the notation $Bas(\Phi)$ will denote $Bas(\mathsf{f}_j)$.

<u>Definition 2.3</u> **F-morphism**

Let M_1 and M_2 be as in definition1.2. *F-morphism*, $I: M_1 \rightarrow M_2$, is a pair of functions $< I_O, I_F >$, where I_O is a basic morphism of M_1 to M_2 and I_F is a founction of the set of formulas of $\mathbf{L}(M_1)$ to the set of the formulas $\mathbf{L}(M_2)$, which is defined as follows.

For any formula Φ of $\mathbf{L}(M_1)$:

(1) If Φ is an atomic formula $\mathsf{P}(\mathsf{t}_1, \ldots, \mathsf{t}_n)$, then

$$I_F(\Phi) = \begin{cases} Bas(\Phi)(\mathsf{t}_1, \ldots, \mathsf{t}_n), & \text{if } T(\mathsf{P}) \text{ is the empty set} \\ (\exists \mathsf{x}_{\mathsf{k}_1} \ldots \mathsf{x}_{\mathsf{k}_m})(Bas(\Phi)(\mathsf{x}_1, \ldots, \mathsf{x}_n) & \\ \quad \wedge (\bigwedge (I_F(\mathsf{x}_{\mathsf{k}_i} = \mathsf{t}_{\mathsf{k}_i}) \mid \mathsf{t}_{\mathsf{k}_i} \in T(\mathsf{P})))), & \text{if } T(\mathsf{P}) = \{\mathsf{t}_{\mathsf{k}_1}, \ldots, \mathsf{t}_{\mathsf{k}_m}\} \end{cases}$$

where every x_{k_i} is a variable not occuring in P and

$$x_i = \begin{cases} x_{k_p}, & \text{if } t_i = t_{k_p} \in \mathcal{T}(P) \\ t_i, & \text{otherwise.} \end{cases}$$

(2) Otherwise,

$$I_F(\neg\Phi) = \neg(I_F(\Phi));$$

$$I_F(\Phi_1 \wedge \Phi_2) = (I_F(\Phi_1)) \wedge (I_F(\Phi_2));$$

$$I_F(\forall x\Phi) = (\forall x)(I_F(\Phi)).$$

Since $\mathcal{T}(\Phi)$ eventually becomes empty, I_F is well-defined.

3 Equivalence between a Peti Net Structure and a Finite Automaton

In this section we construct an F-morphism of a given finite automaton structure to a Petri net structure, and show that all the properties holding in the finite automaton also hold in the Petri net.

Definition 3.1 **Finite Automaton Structure**

A finite automaton structure FA is the following system model.

$$FA =< A \cup B \cup C; A, B, C, \phi, \rho >$$

where

$$\begin{aligned} A, B, C &: \text{ unary relations} \\ \phi, \rho &: \text{ binary relations such that} \end{aligned}$$

$$\phi : (A \cup B \cup C) \times (A \cup B \cup C) \to C$$

$$\phi(a, b) = a \text{ if } a \notin C \text{ or } b \notin A,$$

$$\rho : (A \cup B \cup C) \times (A \cup B \cup C) \to B$$

$$\rho(a, b) = a \text{ if } a \notin C \text{ or } b \notin A.$$

The conditions on $a \notin C$ or $b \notin A$ for ϕ and ρ need only to make the functions ϕ and ρ total, since the first order language we use does not allow partial functions. However, since we will restrict the sentences to the extent as defined later, when we describe the properties of system models, we can regard ϕ and ρ intrinsically as $\phi : C \times A \to C$ and $\rho : C \times A \to B$.

Definition 3.2 **Petri Net Structure**

A Petri net PN is the following system model.

$$PN =< P \cup T \cup N; P, T, I, O, \hat{N} >$$

where

$$\begin{array}{rcl} P,T & : & \text{unary relations} \\ N & : & \text{the set of natural numbers} \\ \hat{N} & : & \text{the set of constants corresponding to } N \\ I,O & \subset & P \times T \times N \end{array}$$

P denotes the set of places and T the set of transitions. $I(p,t,n)$ means that there are n arcs from the place p to the transition t. $O(p,t,n)$ has similar meaning.

There are some ways to construct PA which is considered to have equivalent structure to PN.[5] Following Peterson[5] with some modification, we define PN considered as equivalent to FA. Then our aim is to construct an F-morphism between FA and PN, and to show that the constructed PN preserves all the properties satisfied in FA.

Definition 3.3

$$PN =< P \cup T \cup N; P, T, I, O, \hat{N} >$$

where

$$\begin{array}{rcl} P & = & C \cup A \cup B; \\ T & = & \{t_i | i \in C \times A \cup A \cup B\}; \\ I & = & I_1 \cup I_0 \\ & & \text{where} \\ & & I_1 = \left\{ (p, t_i, 1) \left| \begin{array}{l} \text{either } i = (c,a) \in C \times A \text{ and } (p = c \text{ or } a), \\ \text{or } i = p \in B \end{array} \right. \right\} \\ & & I_0 = \{(p, t_i, 0) \mid (p, t_i, 1) \notin I_1, p \in P, t_i \in T\} : \\ O & = & O_1 \cup O_0 \\ & & \text{where} \\ & & O_1 = \left\{ (p, t_i, 1) \left| \begin{array}{l} \text{either } i = (c,a) \in C \times A \text{ and} \\ \qquad (\phi(c,a) = p \text{ or } \rho(c,a) = p), \\ \text{or } i = p = a \in A \end{array} \right. \right\} \\ & & O_0 = \{(p, t_i, 0) \mid (p, t_i, 1) \notin 0_1, p \in P, t_i \in T\}. \end{array}$$

Example. Let $FA =< A \cup B \cup C; A, B, C, \phi, \rho >$, where $A = B = \{0,1\}$ and $C = \{c_1, c_2\}$. Fig.3.1 illustrates the state transition of FA. The graph of the corresponding PN defined in definition 3.3 is depicted in Fig.3.2.

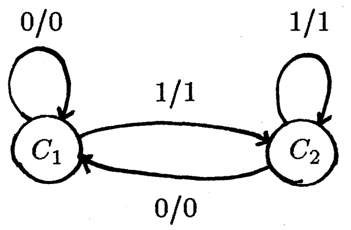

Fig.3.1 An example of a finite automaton

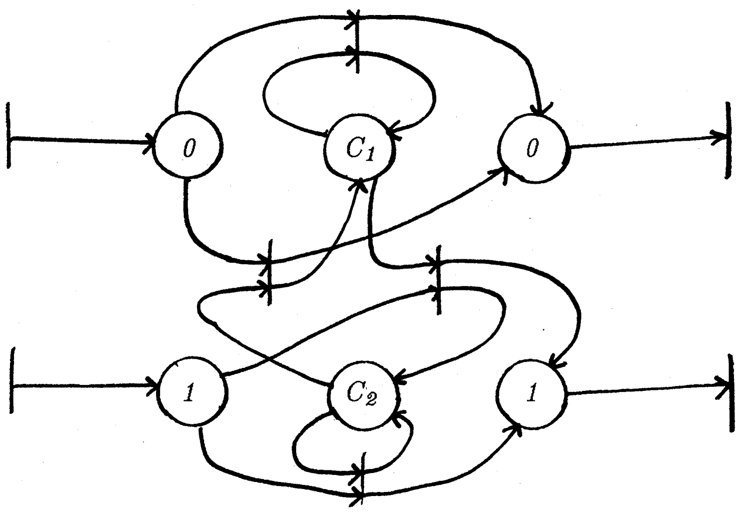

Fig.3.2 An example of a Petri net equivalent to FA

Then we can define an F-morphism between FA and the corresponding PN.

Definition 3.4

Let FA and PN be as in definitions 3.1 and 3.3 respectively. An F-morphism $I = < I_O, I_F >: FA \rightarrow PN$ is defined as follows.

$$
\begin{array}{lcl}
I_O & : & \text{the inclusion map;} \\
I_F(\mathsf{A}(\mathsf{x})) & = & (\mathsf{P}(\mathsf{x}) \wedge (\exists \mathsf{t} \in \mathsf{T})((\forall \mathsf{p} \in \mathsf{P})(\mathsf{I}(\mathsf{p},\mathsf{t},0) \wedge \mathsf{O}(\mathsf{x},\mathsf{t},1)))); \\
I_F(\mathsf{B}(\mathsf{x})) & = & (\mathsf{P}(\mathsf{x}) \wedge (\exists \mathsf{t} \in \mathsf{T})((\forall \mathsf{p} \in \mathsf{P})(\mathsf{O}(\mathsf{p},\mathsf{t},0) \wedge \mathsf{I}(\mathsf{x},\mathsf{t},1)))); \\
I_F(\mathsf{C}(\mathsf{x})) & = & (\mathsf{P}(\mathsf{x}) \wedge \neg I_F(\mathsf{A}(\mathsf{x})) \wedge \neg I_F(\mathsf{B}(\mathsf{x}))); \\
I_F(\phi(\mathsf{x},\mathsf{y}) = \mathsf{z}) & = & ((I_F(\mathsf{C}(\mathsf{x})) \wedge I_F(\mathsf{A}(\mathsf{y})) \rightarrow (\exists \mathsf{t} \in \mathsf{T})(\mathsf{I}(\mathsf{x},\mathsf{t},1) \wedge \mathsf{I}(\mathsf{y},\mathsf{t},1) \wedge \mathsf{O}(\mathsf{z},\mathsf{t},1) \wedge I_F(\mathsf{C}(\mathsf{z})))) \\
 & & \wedge(\neg I_F(\mathsf{C}(\mathsf{x})) \vee \neg I_F(\mathsf{A}(\mathsf{y})) \rightarrow \mathsf{z} = \mathsf{x})); \\
I_F(\rho(\mathsf{x},\mathsf{y}) = \mathsf{z}) & = & ((I_F(\mathsf{C}(\mathsf{x})) \wedge I_F(\mathsf{A}(\mathsf{y})) \rightarrow (\exists \mathsf{t} \in \mathsf{T})(\mathsf{I}(\mathsf{x},\mathsf{t},1) \wedge \mathsf{I}(\mathsf{y},\mathsf{t},1) \wedge \mathsf{O}(\mathsf{z},\mathsf{t},1) \wedge I_F(\mathsf{B}(\mathsf{z})))) \\
 & & \wedge(\neg I_F(\mathsf{C}(\mathsf{x})) \vee \neg I_F(\mathsf{A}(\mathsf{y})) \rightarrow \mathsf{z} = \mathsf{x})).
\end{array}
$$

This definition clearly satisfies the condition required for F-morphisms. Also we can see, as the following lemmas show, that the image of the above F-morphism preserves the structure of FA.

Lemma 3.1

$$A = \{a | PN \models I_F(\mathsf{A}(\mathsf{x}))[a], a \in |PN|\};$$

$$B = \{b | PN \models I_F(\mathsf{B}(\mathsf{x}))[b], b \in |PN|\};$$

$$C = \{c | PN \models I_F(\mathsf{C}(\mathsf{x}))[c], c \in |PN|\}.$$

Lemma 3.2

$$\{(a_1, a_2, a_3) | FA \models (\phi(\mathsf{x}, \mathsf{y}) = \mathsf{z})[a_1, a_2, a_3]\}$$

$$= \{(a'_1, a'_2, a'_3) | PN \models Bas(\phi)(\mathsf{x}, \mathsf{y}, \mathsf{z})[a'_1, a'_2, a'_3], a'_1, a'_2, a'_3 \in P\};$$

$$\{(b_1, b_2, b_3) | FA \models (\rho(\mathsf{x}, \mathsf{y}) = \mathsf{z})[b_1, b_2, b_3]\}$$

$$= \{(b'_1, b'_2, b'_3) | PN \models Bas(\rho)(\mathsf{x}, \mathsf{y}, \mathsf{z})[b'_1, b'_2, b'_3], b'_1, b'_2, b'_3 \in P\}.$$

Our aim is to investigate what properties of FA are preserved by I. To this end, we need to define many-sorted sentences.

Definition 3.5 Many-Sorted Formula

Let **L** be a first order language. A *many-sorted formula* is a formula of **L** and defined recursively.

(1) An atomic formula is a many-sorted formula;

(2) if ϕ and ψ are many-sorted formulas, then $(\phi \wedge \psi)$ and $\neg\phi$ are many-sorted formulas;

(3) if Q is a unary relation of **L** and ϕ is many-sorted formula, theb $(\forall \mathsf{x} \in \mathsf{Q})\phi$ is a many-sorted formula.

A many-sorted formula whose variables are all bounded is said to be a *many-sorted sentence.*

Even if we restrict sentences for the description to many-sorted sentences, the ability of the description is not less than that with ordinary first order sentences[6].

The following theorem shows a typical type of equivalence between PN and FA.

Theorem 3.1

Let $I =< I_O, I_F >$ be the F-morphism defined in definition 3.4. Then for any many-sorted sentence Φ of $\mathbf{L}(FA)$,

$$FA \models \Phi \quad \textit{iff} \quad PN \models I_F(\Phi).$$

This theorem implies that the structure of FA is embedded in PN constructed in definition 3.3, and all the properties of FA are preserved there.

We should notice that the dynamic behevior of PN by the transition of marking is implied by the relation O of PN, which can also represent the firing of the transitions.

4 Equivalence of Behavior between a Petri Net and a Finite Automaton

PN formulated in definition 3.2 has no rule on firing. In this section we formulate a Petri net with marking in the way to specify the transition of marking by firing. Then we discuss the equivalence of behavior between a Petri net with marking and a finite automaton. Through the section we deal with only bounded Petri nets.

Definition 4.1 **Petri Net with Marking**

A Petri net with marking PN_m is defined as follows.

$$PN_m = < P \cup T \cup N \cup M; P, T, I, O, \hat{N}, M, \delta >$$

where

$$\begin{array}{rcl} P,T,I,O,N,\hat{N} & : & \text{as in definition 3.2} \\ M & : & \text{a unary relation, } M = \{f | f : P \to N\} \\ \delta & : & \text{a binary function} \\ & & \delta \text{ is an arbitrary extension of } \delta' : M \times T \to M \end{array}$$

M stands for the whole set of the marking of PN_m and δ stands for the transition of the marking by firing.

The behavior of PN_m is all represented by δ. We reduct PN_m to the model $< T, M, \delta >$ which describes δ, and FA to the state-transition system $< X, C, \phi >$. These reductions do not lose the generality of the consideration on the equivalence of behavior.

Let

$$PN_m^\delta = < T \cup M; T, M, \delta >$$

be a reducted submodel of PN_m and

$$FA^\phi = < A \cup C; A, C, \phi >$$

a reducted submodel of FA. Then the following theorem holds.

Theorem 4.1

Let PN_m^δ and FA^ϕ be as above. Then if PN_m is bounded, there exist FA^ϕ and an F-morphism $I : PN_m^\delta \to FA^\phi$ such that

$$PN_m^\delta \cong FA^\phi.$$

This theorem implies that all the sentences about PN_m^δ are preserved in FA^ϕ.

5 Conclusion

In this paper we showed in a completely formal way that a Petri net structure could simulate a finite automaton. The framework we developed could explicitly and rigorously define the description ability of system models by using F-morphisms. This paper provides a "non-trivial" example of F-morphisms which give similarity between models of different types.

References

1. Ichikawa,A.,1984,*Lecture Note on Systems Engineering* (Tokyo Institute of Technology, Dept. of Systems Science).
2. Zadeh,L.A. and Polak,E.,1969,*System Theory* (New York : McGraw-Hill).
3. Takahara,Y and Takahashi,S,1989,A General Framework of Structural Similarity between System Models, (submitted)
4. Bridge,J.,1977,*Beginning Model Theory* (London : Oxford Univ. Press).
5. Peterson,J.L.,1981,*Petri Net Theory and the Modeling of Systems* (Practice-Hall)
6. Enderton,H.B.,1972,*A Mathematical Introduction to Logic* (New York : Academic Press).
7. Takahashi,S.,1989,*Logical Foundations of Structure and Similarity of System Models* (Ph.D dissertation : Tokyo Institute of Technology).

6 Appendix

Proof of Lemma 3.1

We prove $A = \{a | PN \models I_F(\mathsf{A(x)})[a]\}$. Let $a \in A$. Since I_O is the inclusion map, $I_O(a) = a \in P$. From the definition of I and O, if $i \in A$, then $I(p, t_i, 0)$ for any $p \in P$ and $O(a, t_i, 1)$ hold. So we have $PN \models I_F(\mathsf{A(x)})[a]$.

Conversely let $PN \models I_F(\mathsf{A(x)})[a]$. Then

$$P(a) \wedge (\exists t_i \in T)((\forall p \in P)(I(p, t_i, 0) \wedge O(a, t_i, 1)))$$

holds in PN. From $P(a)$, $a \in C \cup A \cup B$. From $(\forall p \in P)(I(p, t_i, 0))$, we have $i \in A$, hence, $a \in A$ by $O(a, t_i, 1)$.

The rest can be similarly proven. □

Proof of Lemma 3.2

Let $FA \models (\phi(\mathsf{x},\mathsf{y}) = \mathsf{z})[a_1, a_2, a_3]$. Then $\phi(a_1, a_2) = a_3$.

Case: $a_1 \in C$ and $a_2 \in A$.

From Lemma 3.1, $PN \models I_F(\mathsf{C}(\mathsf{x}) \wedge \mathsf{A}(\mathsf{y}))[a_1, a_2]$. Let $i = (a_1, a_2) \in C \times A$. From definition 3.3, $(a_1, t_i, 1), (a_2, t_i, 1) \in I$ and $(\phi(a_1, a_2), t_i, 1) \in O$. Since $\phi(a_1, a_2) = a_3 \in C$ from the definition of FA, we have $PN \models Bas(\phi)(\mathsf{x},\mathsf{y},\mathsf{z})[a_1, a_2, a_3]$ by definition 3.4.

Case: $a_1 \notin C$ or $a_2 \notin A$.

From the definition of FA, $a_1 = a_3$. And from Lemma 3.1, $PN \models \neg I_F(\mathsf{C}(\mathsf{x}) \wedge \mathsf{A}(\mathsf{y}))[a_1, a_2]$. So we have $PN \models Bas(\phi)(\mathsf{x},\mathsf{y},\mathsf{z})[a_1, a_2, a_3]$.

Conversely let $PN \models Bas(\phi)(\mathsf{x},\mathsf{y},\mathsf{z})[a_1', a_2', a_3']$, where $a_1', a_2', a_3' \in P$.

Case: $a_1' \in C$ and $a_2' \in A$.

Then

$$PN \models (\exists t \in \mathsf{T})(\mathsf{I}(\mathsf{x},\mathsf{t},1) \wedge \mathsf{I}(\mathsf{y},\mathsf{t},1) \wedge \mathsf{O}(\mathsf{z},\mathsf{t},1) \wedge I_F(\mathsf{C}(\mathsf{z})))[a_1', a_2', a_3']$$

From the definition of I, $t_i \in T$ for $i = (a_1', a_2')$ satisfies the above formula. Due to $a_3' \in C$ and the definition of O, $O(a_3', t_i, 1)$ means $\phi(a_1', a_2') = a_3'$. So we have $FA \models (\phi(\mathsf{x},\mathsf{y}) = \mathsf{z})[a_1', a_2', a_3']$.

Case: $a_1' \notin C$ or $a_2' \notin A$.

Then

$$PN \models (\neg \mathsf{C}(\mathsf{x}) \vee \neg \mathsf{A}(\mathsf{y}) \to \mathsf{z} = \mathsf{x})[a_1', a_2', a_3'].$$

So we have $a_1' = a_3'$. From the definition of FA, we have $\phi(a_1', a_2') = a_1' = a_3'$. Hence $FA \models (\phi(\mathsf{x},\mathsf{y}) = \mathsf{z})[a_1', a_2', a_3']$.

The rest can be similarly proven. $\square$

<u>Proof of Theorem 3.1</u>

First we define the extended models of FA and PN obtained by adjoining to FA and PN a new constant for each element in $C \cup A \cup B$ as follows.

$$< FA, U > = < C \cup A \cup B; A, B, C, \phi, \rho, U >,$$

$$< PN, U > = < P \cup T \cup N; P, T, I, O, \hat{N}, U >,$$

where $< C \cup A \cup B; A, B, C, \phi, \rho >$ and $< P \cup T \cup N; P, T, I, O, \hat{N} >$ are the underlying models FA and PN respectively, and U is the set of new constants.

Since FA and PN are reducts of $< FA, U >$ and $< PN, U >$ respectively, each many-sorted sentence of FA (or PN) is also a many-sorted sentence of $< FA, U >$ (or $< PN, U >$). Then for any many-sorted sentence Φ of FA we have

$$FA \models \Phi \text{ iff } < FA, U > \models \Phi$$

and

$$PN \models I_F(\Phi) \text{ iff } < PN, U > \models I_F^*(\Phi),$$

where the basic morphism Bas^* of I_F^* is an extension of Bas, that is:

$$Bas^*(P) = \begin{cases} Bas(P) & \text{if } P \in \mathbf{L}(FA) \\ P & \text{if } P \in U \end{cases}$$

So it is sufficient to show that for any many-sorted sentence Φ of $< FA, U >$

$$< FA, U > \models \Phi \text{ iff } < PN, U > \models I_F^*(\Phi).$$

We show this by induction on the length of Φ.

(1) Φ is an atomic sentence.

Let Φ be $\mathsf{R_i}(t_1, \ldots, t_{\lambda(i)})$, where $t_1, \ldots, t_{\lambda(i)}$ are closed terms. We should notice that $\mathsf{R_i}$ is A or B or C or $=$, and so $\lambda(i) \leq 2$. We define $F_n(P)$ for a formula P to indicate the number of the function symbols other than the constants occuring in P. We show the claim in the case of atomic sentences by induction on $F_n(\Phi)$. The case where $\mathsf{R_i}$ is $=$ can be ascribed to other cases.

$F_n(\Phi) = 0$.

$\mathsf{R_i}$ must be A or B or C, and $t \in U$. We first show the only if part. Suppose $< FA, U > \models \mathsf{R_i}(t)$. By the definition, $I_F^*(\mathsf{R_i}(t)) = (\exists \mathsf{x})(Bas^*(\mathsf{R_i})(\mathsf{x}) \wedge I_F^*(\mathsf{x} = t))$. Since $t \in U$, we have $I_F^*(\mathsf{x} = t) = (\mathsf{x} = t)$. Let t^d be the interpretation of t in $< FA, U >$ and $< PN, U >$. Then $t^d \in R_i \subset P$. Therefore $< FA, U > \models (\mathsf{R_i}(\mathsf{x}) \wedge (\mathsf{x} = t))[t^d]$. Since, by Lemma 3.1, $Bas^*(\mathsf{R_i})(t^d)$ holds in $< PN, U >$, we have $< PN, U > \models (Bas^*(\mathsf{R_i})(\mathsf{x}) \wedge I_F^*(\mathsf{x} = t))[t^d]$, and so $< PN, U > \models I_F^*(\mathsf{R_i}(t))$.

Conversely suppose $< PN, U > \models I_F^*(\mathsf{R_i}(t))$. By the definition of I_F^*, there exists an interpretation t^d of t in $\mathsf{C} \cup \mathsf{A} \cup \mathsf{B}$ such that $Bas^*(\mathsf{R_i})(t^d)$ holds in $< PN, U >$. By Lemma 3.1, $R_i(t^d)$ holds in $< FA, U >$. Therefore $< FA, U > \models (\exists \mathsf{x})(\mathsf{R_i}(\mathsf{x}) \wedge (\mathsf{x} = t))$.

$F_n(\Phi) = 1$.

We devide this case into two cases.

Case 1: $\mathsf{R_i}$ is ϕ or ρ.

We denote ϕ or ρ by a symbol f. Then we have to show $< FA, U > \models (f(t_1, t_2) = t_3)$ iff $< PN, U > \models I_F^*(f(t_1, t_2) = t_3)$ for any $t_1, t_2, t_3 \in U$. By the definition of I_F^* and Lemma 3.2, this is clear.

Case 2: $\mathsf{R_i}$ is A or B or C.

Then we can write Φ as $\mathsf{R_i}(f(t_1, t_2))$, where f is ϕ or ρ and $t_1, t_2 \in U$.

$$\begin{aligned} < FA, U > \models \mathsf{R_i}(f(t_1, t_2)) \quad & \text{iff} \quad < FA, U > \models (\mathsf{R_i}(t_3) \wedge t_3 = f(t_1, t_2)) \\ & \qquad \text{for some } t_3 \in U \\ & \text{iff} \quad < PN, U > \models (Bas^*(\mathsf{R_i})(t_3) \wedge I_F^*(t_3 = f(t_1, t_2)) \\ & \text{iff} \quad < PN, U > \models (\exists \mathsf{x})(Bas^*(\mathsf{R_i})(\mathsf{x}) \wedge I_F^*(\mathsf{x} = f(t_1, t_2)) \\ & \text{iff} \quad < PN, U > \models I_F^*(\mathsf{R_i}(f(t_1, t_2))). \end{aligned}$$

Suppose it holds for $F_n(\Phi) < k$. We prove it for $F_n(\Phi) = k$.

For the simplicity of notation, but without loss of generality, we consider a unary relation R. Let t be a closed term of FA.

Then

$$< FA, U > \models \mathsf{R}(t) \text{ iff } < FA, U > \models (\mathsf{R}(u) \wedge u = t),$$

where $u \in U$ and $u^d = t^d$ (i.e. the interpretations of u and t are equivelent). On the other hand,

$$\begin{aligned} <PN,U> \models I_F^*(\mathsf{R}(t)) \quad &\text{iff} \quad <PN,U> \models (\exists \mathsf{x})(I_F^*(\mathsf{R}(u)) \wedge I_F^*(x = t)) \\ &\text{iff} \quad <PN,U> \models I_F^*(\mathsf{R}(u) \wedge I_F^*(u = t)) \\ &\text{iff} \quad <PN,U> \models I_F^*(\mathsf{R}(u)) \text{ and} \\ &\qquad <PN,U> \models I_F^*(u = t). \end{aligned}$$

By the induction hypothesis, we have

$$<FA,U> \models \mathsf{R}(u) \text{ iff } <PN,U> \models I_F^*(\mathsf{R}(u)).$$

So it sufficies to show that

$$<FA,U> \models (u = t) \text{ iff } <PN,U> \models I_F^*(u = t).$$

Let t be $\mathsf{f_j}(s_1, \ldots, s_{\mu(j)})$.

Then $<FA,U> \models (u = t)$ iff there exist $c_{k_1}, \ldots, c_{k_m} \in U$ such that

$$<FA,U> \models (u = \mathsf{f_j}(s'_1, \ldots, s'_{\mu(j)}) \wedge (\bigwedge(c_{k_i} = s_{k_i} | s_{k_i} \in T(\mathsf{f_j})))),$$

where

$$s'_i = \begin{cases} c_{k_p} & \text{if } s_i = s_{k_p} \in T(\mathsf{f_j}))) \\ s_i & \text{otherwise} \end{cases}$$

iff there exist $c_{k_1}, \ldots, c_{k_m} \in U$ such that

$$<PN,U> \models (I_F^*(u = \mathsf{f_j}(s'_1, \ldots, s'_{\mu(j)})) \wedge (\bigwedge I_F^*(c_{k_i} = s_{k_i} | s_{k_i} \in T(\mathsf{f_j}))))$$

(by the induction hypothesis),

iff $<PN,U> \models (\exists \mathsf{x_{k_1}}, \ldots, \mathsf{x_{k_m}})(u = \mathsf{f_j}(s''_1, \ldots, s''_{\mu(j)}) \wedge (\bigwedge(I_F^*(\mathsf{x_{k_i}} = s_{k_i}) | s_{k_i} \in T(\mathsf{f_j}))),$ where

$$s''_i = \begin{cases} \mathsf{x_{k_p}} & \text{if } s'_i = c_{k_p} \in T(\mathsf{f_j}))) \\ s_i & \text{otherwise} \end{cases}$$

(Notice that the constants occuring in t are only in U.)

iff $<PN,U> \models I_F^*(u = t)$.

(2) Φ is $\psi_1 \wedge \psi_2$.

$$\begin{aligned} <FA,U> \models \Phi \quad &\text{iff} \quad <FA,U> \models \psi_1 \text{ and } <FA,U> \models \psi_2 \\ &\text{iff} \quad <PN,U> \models I_F^*(\psi_1) \text{ and } <PN,U> \models I_F^*(\psi_2) \\ &\text{iff} \quad <PN,U> \models I_F^*(\psi_1 \wedge \psi_2) \\ &\text{iff} \quad <PN,U> \models I_F^*(\Phi). \end{aligned}$$

(3) Φ is $\neg\psi$.

$$\begin{aligned} <FA,U> \models \Phi \quad &\text{iff} \quad \text{not } <FA,U> \models \psi \\ &\text{iff} \quad \text{not } <PN,U> \models I_F^*(\psi) \\ &\text{iff} \quad <PN,U> \models I_F^*(\neg\psi) \\ &\text{iff} \quad <PN,U> \models I_F^*(\Phi) \end{aligned}$$

(4) Φ is $(\forall \mathsf{x} \in \mathsf{R})\psi(\mathsf{x})$ where R is a unary relation symbol.
Let t be a constant in U whose interpretation t^d is in R.

$$
\begin{array}{rll}
< FA, U > \models \Phi & \text{iff} & \text{for any } a \in R, < FA, U > \models (\psi(\mathsf{x}))[a] \\
& \text{iff} & \text{for any } t \in U, \text{ where } t^d = a \in R, \\
& & \quad < FA, U > \models \psi(t) \\
& \text{iff} & \text{for any } t \in U \ (t^d \in R), < PN, U > \models I_F^*(\psi(t)) \\
& & \qquad \text{(by the induction hypothesis)} \\
& \text{iff} & \text{for any } t^d \in R, < PN, U > \models I_F^*(\psi(\mathsf{x}))[t^d] \\
& \text{iff} & \text{for any } t^d \in \{a | PN \models I_F(\mathsf{R}(\mathsf{x}))[a]\}, \\
& & \quad < PN, U > \models I_F^*(\psi(\mathsf{x}))[t^d] \\
& & \qquad \text{(by Lemma 3.1)} \\
& \text{iff} & < PN, U > \models (\forall \mathsf{x})(I_F^*(\mathsf{R}(\mathsf{x})) \rightarrow I_F^*(\psi(\mathsf{x}))) \\
& & \qquad\qquad (I_F^*(\mathsf{R}(\mathsf{x})) = I_F(\mathsf{R}(\mathsf{x})).) \\
& \text{iff} & < PN, U > \models I_F^*(\Phi),
\end{array}
$$

which completes the proof. □