National Library of Canada

Acquisitions and Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale du Canada

Direction des acquisitions et des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Canada

Invariant-Preserving Transformations for the Verification of Place/Transition Systems

(with application to the verification of protocols)

by

Wei Zeng

A M.Sc. Thesis

Submitted to the School of Graduate Studies and Research
in partial fulfillment of the requirements for the
Master of Computer Science Degree*

University of Ottawa
Ottawa, Ontario
Canada

July 1994

*The Master of Computer Science Program is a joint program with Carleton University, administered by the Ottawa-Carleton Institute for Computer Science

ISBN 0-612-07813-2

Canada

UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

ABSTRACT

Transformations preserving specific properties are often used for the verification of place/transition systems. They simplify a system so that some designated properties can be detected more easily from the simplified system but are still valid for the original one.

This thesis presents five general classes of transformations on place/transition systems (PTSs), namely, insertion, elimination, replacement, composition and decomposition, and the conditions for them to preserve place-invariants and transition-invariants of the PTSs. Also proposed is a special transformation for eliminating isolatable places. It leads to the proposal of a constructive algorithm for finding place-invariants. The algorithm simplifies a PTS iteratively in such a way that its place-invariants can be derived from those of the simplified ones.

Lastly, some of these transformations and the algorithm are applied to find place-invariants for two classes of the Transport Protocol.

Key words: Composition, decomposition, elimination, insertion, invariant, Petri net, place/transition system, preserve, protocol, replacement, transformation, verification.

ACKNOWLEDGMENT

I would like to express my deepest appreciation to my thesis supervisor Professor To-yat Cheung for his valuable time, patience, guidance and advice throughout my graduate studies. His instruction for revising the drafts of the thesis has greatly improved its contents and presentation. The many discussions we have made have been of great influence and assistance.

I acknowledge with gratitude the financial supports provided by City Polytechnic of Hong Kong, Natural Sciences and Engineering Research Council of Canada and the Telecommunications Research Institute of Ontario, Canada.

I am also indebted to the special supports from my friend Z. Dai, my brothers Ping, Xiao-jing and Guang, and especially from my parents Guo-liang and Xue-qin.

TABLE OF CONTENTS

Abstract i

Acknowledgment ii

Table of Contents iii

List of Figures v

Chapter 1 INTRODUCTION AND FUNDAMENTALS 1

1.1 Specification, Verification and Property-preserving Transformations 1

1.2 Motivation and Contributions of the Thesis 3

1.3 Outline of the Thesis 5

1.4 Place/Transition Systems (PTS's) and Their Properties 6

Chapter 2 REVIEW ON PROPERTY-PRESERVING TRANSFORMATIONS ON PTS's 11

2.1 Lee's Transformations [LEE85, LEE87] 11

2.2 Berthelot's Transformations [BER84, BER87] 15

2.3 Ramamoorthy's Transformations [RAM86] 18

2.4 Other Transformations 20

Chapter 3 INSERTION AND ELIMINATION 22

3.1 Insertion-I and Its Matrix Representation with Vertical Vectors 22

3.2 Conditions for Insertion-I to Preserve Place-Invariants 23

3.3 Matrix Representation of Insertion-I with Horizontal Vectors 26

3.4 Conditions for Insertion-I to Preserve Transition-Invariants 27

Chapter 4 ELIMINATING ISOLATABLE PLACES 28

4.1 Isolatable, Isolated and Non-isolatable places 28

4.2 Elimination EIP 29

4.3 Algorithm FPI 31

4.4 An Example 34

Chapter 5 REPLACING PLACES OR TRANSITIONS 35

5.1 Replacement RP and Its Matrix Representation with Vertical Vectors 35

5.2 Conditions for Replacement RP to Preserve Invariants 37

5.3 Replacement RT and Its Matrix Representation with Horizontal Vectors ... 40

5.4 Conditions for Replacement RT to Preserve Invariants 41

Chapter 6 COMPOSITION AND DECOMPOSITION 44

6.1 Composition CP and Its Matrix Representation with Vertical Vectors 44

6.2 Conditions for Composition CP to Preserve Invariants 45

6.3 Composition CT and Its Matrix Representation with Horizontal Vectors ... 47

6.4 Conditions for Composition CT to Preserve Invariants 48

Chapter 7 FINDING PLACE-INVARIANTS OF THE TRANSPORT PROTOCOL BY REDUCTION 49

7.1 The Individual Transport Entities 49

7.2 The Communicating Transport Entities 51

7.3 Detection of Place-Invariants 51

Chapter 8 CONCLUSION AND FUTURE STUDIES 53

References 55

LIST OF FIGURES

Figure 2.1 The sequence t-p-t_o is replaced with a single transition t^* 13

Figure 2.2 The sequence p-t-p_o is replaced with a single place p^* 15

Figure 2.3 Two redundant places, their tokens and connecting arcs are eliminated... 17

Figure 2.4 Fusing two doubled places into a single place..... 18

Figure 2.5 A FWBM is replaced with a single transition t^* 20

Figure 3.1 Representation of Insertion-I in vertical format..... 22

Figure 3.2 A set of places, transitions and (dotted) arcs are inserted into a PTS..... 25

Figure 3.3 Insertion-I in horizontal format..... 26

Figure 4.1 Isolatable place, isolated place and non-isolatable place..... 28

Figure 4.2 Conceptual tree structure of Algorithm FPI..... 33

Figure 4.3 To find a place-invariant by eliminating isolated-places..... 34

Figure 5.1 Representation for Replacement RP in vertical format..... 35

Figure 5.2 Places $\{p_2, p_3, p_4\}$ are replaced with places $\{p_2', p_3'\}$..... 36

Figure 5.3 A Replacement RP is a combination of an Elimination-E and an Insertion-I 36

Figure 5.4 Replacing a set of places while preserving a place-invariant..... 38

Figure 5.5 Merging a subset of places while preserving transition-invariants..... 39

Figure 5.6 Representation of Replacement RT in horizontal vectors..... 40

Figure 5.7 Replacing a set of transitions while preserving transition-invariants..... 42

Figure 5.8 Merging a subset of transitions while preserving place-invariants..... 43

Figure 6.1 Connecting two PTSs by fusing some of their places..... 45

Figure 6.2 PTS1 and PTS2 are connected by fusing one of their places..... 47

Figure 6.3 Connecting two PTSs by fusing some of their transitions..... 48

Figure 7.1 Connection and disconnection phases of two Transport entities..... 50

Figure 7.2 Global net model of two communicating Transport entities..... 51

Figure 7.3 Reduced nets of the individual Transport entities..... 52

Chapter 1

INTRODUCTION AND FUNDAMENTALS

1.1 SPECIFICATION, VERIFICATION AND PROPERTY-PRESERVING TRANSFORMATIONS

Requirement analysis, design, implementation and maintenance are the four main phases in the development of a software system.

In the design phase, a specification of the system is produced. Some means for specification include: specification languages [BER88, BUD87, CCI92, CHE88, ISO89a, ISO89b], temporal logic [BRA90, HAI83, SCH81] and behavior models [BOC78, BRA83, BRA87, DIA87, HOA78, HOA81, HOA85, MIL80, OBA87, PET77, SHA90, WAN91]. In particular, the following three formal languages have been standardized by Consultative Committee of International Telegraphy and Telephony (CCITT) and International Organization for Standardization (ISO): Specification and Description Language (SDL) [CCI92], Extended State-Transition-based Language (ESTELL) [ISO89a] and Language Of Temporal Ordering Specification (LOTOS) [ISO89b]. The following five behavior models have been widely applied to specify various systems in many areas of applications: finite state machine (FSM), labeled transition systems (LTS), calculus of communicating systems (CCS), communicating sequential processes (CSP) and place/transition systems (PTS).

Verification is the process of determining the correctness of a design. One of the main approaches for verification is analysis. In analysis, one checks whether a specification satisfies certain properties. Three widely-used analysis techniques are: program proving, reachability analysis and invariant analysis. Among the many methods for program proving, one is to use logic formulas to reflect the behavior of the system and logical assertions to state the specified properties. The latter are then proved by deductively using

inference rules of the underlying logic. A particular case is to use temporal logic for proving the liveness and other properties of a distributed system [APT85, BRO83]. Reachability analysis is based on generating and exploring the states of the system in the form of a tree. Certain properties can be detected during the exploration. Variations of reachability analysis can be found in [BIL88, CHE89, DAN77, GOU83, ITO83, LAM84, LIN87, SHA90, YUA89]. In invariant analysis, those properties of a system which remain unchanged during execution can be detected. Many properties, such as liveness, boundedness and home states, can be checked by analyzing the place-invariants and transition-invariants of its Petri-net representation [BER82, REI85]. A commonly-used method for invariant analysis is by solving a set of integer equations derived from a Petri-net.

As systems become more large and complicated, their verification by these three techniques is getting more difficult and lengthy. The number of logical formulas in program proving, the number of states in reachability analysis and the number of equations in invariant analysis may become extremely large. Normal procedures of checking and solution become very time-consuming. Also, in order to verify a property P based on a PTS, it is sometimes very difficult to work directly on the PTS itself and much easier to work indirectly on another system PTS' derived from PTS. We need transformations which either reduce the complication of the system or make the systems reveal the desirable properties more easily. Naturally, in order for this approach to be correct, a transformation should at least satisfy the condition that P exists in PTS if and only if it exists in PTS'. That is, the transformation *preserves property P*. In general, the meaning of 'preserving a property P' should be understood in a general sense and is problem-dependent. In particular, it should not be rigidly quantified. For example, 'a system preserves a place-invariant' does not necessarily mean that the invariant possesses 'the same set of places' before and after the transformation. After a transformation, the set of places should be allowed to increase or decrease.

An example of a transformation on the reachability-tree generated for a system consisting of two communicating entities can be taken from [GOU83]. It generates the reachable states by alternatively allowing a maximal progress to each of the two entites. This transformation is nonprogress-state-preserving. It has been proved that a given system cannot reach a nonprogress state iff none of the states generated in each subtask is a nonprogress state.

Many property-preserving transformations have been investigated for the specification and verification of protocols in recent years [BER84, BER87, CIN85, LEE85, LEE87, RAM86, REI92]. They either replace special subnet patterns with single places or transitions [BER84, BER87, LEE85, LEE87, RAM86], or adding (resp., deleting) special places and arcs into (resp., from) the nets [BER84, BER87, CIN85]. These transformations are not very general. The conditions for them to preserve properties may be derived from the structure or from the set (or subset) of the reachable states of the system.

1.2 MOTIVATION AND CONTRIBUTIONS OF THE THESIS

In this thesis, we propose five general classes of transformations and the conditions for them to preserve place-invariants or transition-invariants. Our research is motivated by the following facts:

a. The property-preserving transformations existing in the literature are quite restrictive in the sense that they either eliminate a specific subnet pattern, or replace a specific subnet pattern with a single place or transition. In order to make the transformations more useful, it is important to investigate more general transformations.

b. Place-invariants and transition-invariants are important properties which require careful checking. An example of its application is: suppose a PTS contains three places representing two buffers and one channel. If these three places form a

place-invariant with total number of tokens equal to 1, then there always exists one and only one message in these three components of the system.

c. Place-invariants and transition-invariants are useful for proving other properties, such as home-states, liveness and boundedness [BER82, REI85]. For example, if all places in a PTS form a place-invariant, then it can be claimed that the PTS is bounded since no place in the PTS can hold more then a certain number of tokens.

d. At present, for the case of Petri-nets, place-invariants and transition-invariants are detected by obtaining the non-negative integer solutions of a set of linear integer equations derived from the nets themselves. As far as we know, there is no efficient approach for obtaining such solutions. We propose a totally different approach for detecting place-invariants.

This thesis includes two main contributions:

I. We propose five general classes of transformations and the conditions for them to preserve place-invariants or transition-invariants. As described below, these transformations are insertion, elimination, replacement, composition and decomposition:

- An insertion adds a set of places, transitions and arcs into a PTS.
- An elimination deletes a set of places, transitions and arcs from a PTS.
- A replacement substitutes a set of places (resp., transitions) and arcs of a PTS with another set of places (resp., transitions) and arcs.
- A composition fuses a set of places (resp., transitions) of PTS1 and a set of places (resp., transitions) of PTS2 in a one-to-one manner without fusing any transitions (resp., places).

- A decomposition splits a PTS into PTS1 and PTS2 such that PTS1 and PTS2 have a common subset of places (resp., transitions) of PTS without any common subset of transitions (resp., places) of PTS.

II. Based on a special place-invariant-preserving elimination, we propose a constructive algorithm for detecting place-invariants of a PTS.

Besides the numerous examples showing the applications of each transformation, the proposed composition and algorithm have been applied to find place-invariants of the Transport Protocol.

1.3 OUTLINE OF THE THESIS

In order to review the existing methods and to derive the new methods, some basic definitions and terminology of place/transition systems and their properties are first presented in the next section of this chapter.

In Chapter 2, some property-preserving transformations on place/transition systems existing in the literature are reviewed. The properties preserved under these transformations include liveness, deadness, boundedness, unboundedness and exihibited-behavior equivalence.

In Chapter 3, two of the five classes of transformations proposed by us, namely, insertions and eliminations, and conditions for them to preserve place-invariants and transition-invariants are presented. In an insertion transforming PTS to PTS', a set of places (resp., transitions) INV in PTS is a place-invariant (resp., transition-invariant) of PTS iff INV $\cup$ I form a place-invariant (resp., transition-invariant) of PTS', where I is the set of inserted places (resp., inserted transitions), provided that some conditions are satisfied.

In Chapter 4, a special elimination is presented. It deletes a single place, a set of arcs and transitions while preserving place-invariants. This elimination leads to the proposal of a constructive algorithm for finding place-invariants.

In Chapter 5, another class of transformations on a PTS, called replacement and the conditions for it to preserve invariants are proposed. In particular, two kinds of replacements are considered. The first kind replaces a set of places and arcs of PTS with another set of places and arcs. The second kind replaces a set of transitions and arcs of PTS with another set of transitions and arcs.

Chapter 6 presents two more classes of transformations on PTSs, namely composition and decomposition, and the conditions for them to preserve place-invariants and transition-invariants. Two different compositions are considered. One connects two PTSs by fusing some of their places in a one-to-one manner without fusing any of their transitions. Another connects two PTSs by fusing some of their transitions without fusing any of their places. It will be proved that when two PTSs are combined by fusing some of their places, every pair of their place-invariants containing the fused places form a place-invariant for the composite system.

In Chapter 7, the proposed composition (Chapter 6) and the algorithm (Chapter 4) are applied to find a place-invariant for the place/transition system of the first two classes of the ISO Transport Protocol.

In Chapter 8, a short conclusion and future studies are presented.

1.4 PLACE/TRANSITION SYSTEMS (PTS's) AND THEIR PROPERTIES

A place/transition system is a powerful model for representing concurrent behaviors of distributed systems. A finite state machine contains only one kind of nodes, called states. For every arc of a finite state machine, there is a label describing a pair of actions of the system. Unlike a finite state machine, a place/transition system contains two kinds of nodes, namely places and transitions, and there may also be labels on the arcs. A finite

state machine may be considered as a special case of a place/transition system. The formal definitions of a place/transition system and some of its properties are given below.

Definition 1.1. (place/transition system)

A *place/transition system* is a 5-tuple PTS = <S, T, F, W, M_o>, where S is a set of places, T is a set of transitions such that $S \cap T = \emptyset$ and $S \cup T \neq \emptyset$, $F \subseteq (S \times T) \cup (T \times S)$ is the flow set, W is a weight function such that $W(x, y) \in \{1, 2, ...\}$ if $(x, y) \in F$ and $W(x, y) = 0$ if $(x, y) \notin F$. $M_o: S \rightarrow \{0, 1, 2, ...\}$ is the initial marking function.

Graphically, a place/transition system (PTS) consists of two types of nodes: circle nodes representing places, and bar nodes representing transitions. Directed arcs represented by elements of F connect nodes of different types, but not nodes of the same type.

In an application, places and transitions usually play different roles. A transition represents an action, whereas its input and output places represent the preconditions of firability and postconditions after firing of the action. Fulfillment of a precondition represented by a place is indicated by the availability of enough tokens in that place. After the transition is fired, tokens will be removed from each of its input places and added to each of its output places. The markings, firing rule and nets of a PTS are presented below.

Definition 1.2. (marking, firing rule and net)

For a PTS, a *marking* is a function $M: S \rightarrow \{0, 1, 2, ...\}$ such that M(s) represents the number of tokens at place $s \in S$. A transition $t \in T$ is *firable* at M iff $M(s) \geq W(s, t)$ $\forall s \in S$. *Firing* transition t will result in changing marking M to marking M', in notation $M [t> M'$, where $M'(s) = M(s) - W(s, t) + W(t, s)$, $\forall s \in S$. The *net* of a PTS is obtained by ignoring its tokens. When there is no ambiguity, we use the terms 'PTS' and 'the net of PTS' interchangeably.

Definition 1.3. (internal place and internal transition)

For $P \subset S$ and $Q \subset T$, the set of *internal places* of Q w.r.t. PTS and the set of *internal transitions* of P w.r.t. PTS are defined as follows:

$$IP(Q) = \{ p \mid p \in S, \exists t_1, t_2 \in Q: (t_1, p), (p, t_2) \in F\}$$

$$IT(P) = \{ t \mid t \in T, \exists p_1, p_2 \in P: (p_1, t), (t, p_2) \in F\}$$

Definition 1.4. (input set and output set)

For $X \subseteq Y$, where Y is either S or T, the *input set* and *output set* of X w.r.t. Y, in notation $X^\bullet$ and $^\bullet X$, respectively, are defined as follows:

$$^\bullet X = \{ y \mid \exists x \in X, y \in Y: (y, x) \in F\}$$

$$X^\bullet = \{ y \mid \exists x \in X, y \in Y: (x, y) \in F\}.$$

Definition 1.5. (incoming flow, outgoing flow and net flow)

Let N be a set of places (resp., transitions) and N' be a set of transitions (resp., places) of PTS. The *incoming flow*, *outgoing flow* and *net flow* of N w.r.t. N' are defined as follows:

$$IF(N, N') = \sum_{x \in (N' \times N) \cap F} W(x),$$

$$OF(N, N') = \sum_{x \in (N \times N') \cap F} W(x),$$

$$NF(N, N') = IF(N, N') - OF(N, N').$$

The incoming flow is the total weight of those arcs from N' to N. The outgoing flow is the total weight of those arcs from N to N'. The net flow is the difference between the incoming flow and outgoing flow.

For the rest of this thesis, it is assumed that a PTS has m places and n transitions.

Definition 1.6. (pre-incidence matrix, post-incidence matrix and incidence matrix)

The *pre-incidence matrix* PRE of a PTS is an $m \times n$ matrix whose element pre_{ij} is the weight of the arc from place p_i to transition t_j. The *post-incidence matrix* POST of PTS is an $m \times n$ matrix whose element $post_{ij}$ is the weight of the arc from transition t_j to place p_i. V = POST - PRE is called the *incidence matrix* of PTS.

Throughout this thesis, for the sake of convenience, a vector is most of the time written horizontally, though possibly being be vertical or horizontal.

A place-invariant of PTS is a set of places the total number of whose tokens is constant for any execution and any initial marking. It can be represented by a 0/1 m-vector P, such that the subset of places with an element equal to 1 in P forms a place-invariant of PTS. It is well known that P satisfies the equation PV = 0.

Definition 1.7. (place-vector P and place-invariant)

A *place-vector* P is a 0/1 m-vector which satisfies PV = 0. Those places with 1 in P form a *place-invariant* of PTS.

A transition-invariant of a PTS is a set of transitions such that the PTS remains the same marking after the firing of the transitions of this set in a correct sequence [REI85]. However, every transition in this set is not always firable. A transition-vector T is a n-vector, such that the value of ith element of T represents the number of times transition t_i participates in a transition-invariant.

Definition 1.8. (transition-vector T and transition-invariant)

A *transition-vector* T is a n-vector which satisfies VT = 0. A *transition-invariant* of PTS is a set of transitions such that the number of times different transition t in the set is equal to the value of element t in T.

Definition 1.9. (reachability set, liveness, dead and boundedness)

The *reachability set* of PTS, in notation $[M_o>$, is defined as the smallest set of markings of PTS, which satisfies: (a) $M_o \in [M_o>$; and (b) if $\exists M_1 \in [M_o>$ and $t \in T$: $M_1[t>M_2$, then $M_2 \in [M_o>$. PTS is said to be *live* iff $\forall M_1 \in [M_o>$, $\forall t \in T$, $\exists M_2 \in [M_1>$: t is firable at M_2. PTS is said to be dead if $\exists M \in [M_o>$ such that $\forall t \in T$, t is not firable. A place $p \in S$ is said to be *bounded* iff $\exists n \in \{0, 1, 2, ...\}$: $\forall M \in [M_o>$, $M(p) \leq n$. PTS is said to be bounded iff, $\forall p \in S$, p is bounded.

Definition 1.10. (sum, cardinality and inner-product)

For two vectors $\alpha = (a_1, a_2, ..., a_k)$ and $\beta = (b_1, b_2, ..., b_k)$ and a set A, let $I \subseteq \{1, 2, ..., k\}$. The *sum* of α over I is defined as $\{\alpha\}_I = \sum_{i \in I} a_i$. In particular, $\{\alpha\}_I$ can be written as $\{\alpha\}$ if $I = \{1, 2, ..., k\}$. The *cardinality* of α is $|\alpha| = k$, the cardinality of A is |A|, and the inner-product of α and β is $\alpha\beta = \sum_{i=1}^{k} a_i b_i$.

Chapter 2

REVIEW ON PROPERTY-PRESERVING TRANSFORMATIONS ON PTS's

Many property-preserving transformations on PTSs have been studied for the specification and verification of protocols in recent years. In this chapter, the background knowledge and models for some of these transformations and the conditions for them to preserve specific properties are reviewed. Note that this is not a complete review.

2.1 LEE'S TRANSFORMATIONS [LEE85, LEE87]

Twelve reduction methods are studied in [LEE85, LEE87]. Each of them replaces a special subnet with a single place or a single transition. They preserve such behavior properties as liveness, boundedness and proper termination. Based on the structure of the net, these special subnets can be detected. These reduction methods can be automated. To illustrate the main theme of these methods, two of the six transformations in [LEE87] will be described below.

Definition 2.1 (input door, output door and between door)

Let $N = <S_n, T_n, F_n, W_n>$ be a subnet of PTS. The *input door* ID(N), *output door* OD(N) and *between door* BD(N) of N are defined as follows:

$$ID(N) = \{ p \mid p \in S_n, \exists t \in T - T_n: (t, p) \in F\} \cup \{ t \mid t \in T_n, \exists p \in S - S_n: (p, t) \in F\}$$

$$OD(N) = \{ p \mid p \in S_n, \exists t \in T - T_n: (p, t) \in F\} \cup \{ t \mid t \in T_n, \exists p \in S - S_n: (t, p) \in F\}$$

$$BD(N) = S_n \cup T_n - ID(N) \cup OD(N)$$

In Lee's methods, the special subnet patterns have to be detected before a PTS can be transformed. In the following, the definitions of two special subnets, namely *generalized*

reducible subnet-1T and *generalized reducible subnet-1P*, are first presented. Then, the transformations for replacing them with a single place or transition are described.

Definition 2.2 (generalized reducible subnet-1T)

A subnet N of PTS is said to be a *generalized reducible subnet-1T* if it satisfies the following conditions:

a. $ID(N) \subset T$, $OD(N) \subset T$, $BD(N) \subset P$, $|ID(N)| = 1$, $|OD(N)| = 1$ and $|BD(N)| = 1$. Suppose $ID(N) = \{t\}$, $OD(N) = \{t_o\}$ and $BD(N) = \{p\}$.

b. $M(p) = 0$.

c. $t - p - t_o$ is the only path from t to t_o.

d. If $W(t, p) \neq W(p, t_o)$, then either $\exists k \in \{1, 2, ...\}$: $W(t, p) = kW(p, t_o)$ or $\exists k' \in \{1, 2, ...\}$: $W(t, p) = k'W(p, t_o)$.

e. If there exists a circuit which contains t, then $k = k' = 1$.

f. If $k' > 1$, then, $W(t, p) = 0$ $\forall p' \neq p$.

Replacement R-1T (replacing a generalized reducible subnet-1T with a transition t^*)

Let N be a generalized reducible subnet-1T of PTS. If PTS is transformed to PTS' by replacing N with a transition t^*, then PTS is live (resp., bounded) iff PTS' is live (resp., bounded), provided that the following conditions are satisfied:

a. $S' = S - \{p\}$.

b. $T' = T - \{t, t_o\} + \{t^*\}$.

c. $\forall p' \in S - \{p\}$,

$$W'(p',t) = \begin{cases} k'W(p',t) + W(p',t_o), & \text{if } t = t^* \\ W(p',t), & \text{if } t \neq t^* \end{cases}$$

$$W'(t,p') = \begin{cases} kW(t_o,p') + W(t,p'), & \text{if } t = t^* \\ W(t,p'), & \text{if } t \neq t^* \end{cases}$$

$M'(p') = M(p')$. □

This transformation fuses a sequence of alternating transitions and places into a single transition.

Theorem 2.1

Replacement R-1T is correct.

Proof: See [LEE87]. □

Example 2.1 (Figure 2.1) [LEE87]

In PTS, $N = <\{p\}, \{t, t_o\}, \{(t, p), (p, t_o)\}>$ is a generalized reducible subnet-1T such that $W(t,p)/W(p,t_o) = k$. According to Replacement R-1T, PTS can be transformed to PTS' by replacing N with a transition t^* such that, the following conditions are satisfied in PTS': $W'(t^*, p_3) = kW(t_o, p_3)$, $W'(t^*, p_2) = W(t_2, p_2)$ and $W'(p_1, t^*) = W(p_1, t_o)$. As a result, since PTS' is live (resp., bounded), PTS is also live (resp., bounded).

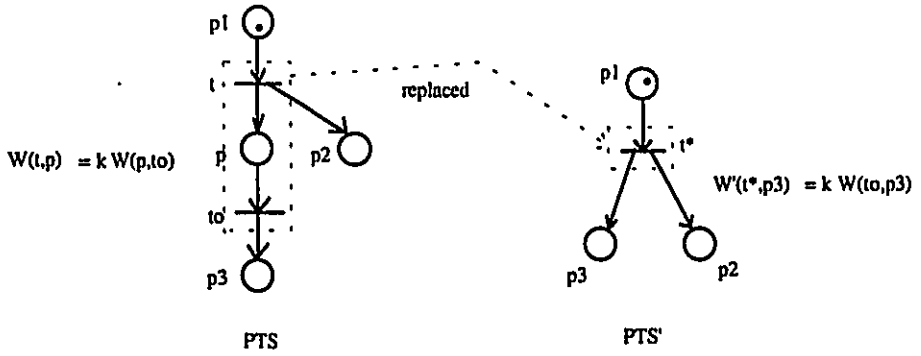


Figure 2.1. The sequence $t - p - t_o$ is replaced with a single transition t^*.

Definition 2.3 (generalized reducible subnet-1P)

A subnet N of PTS is said to be a *generalized reducible subnet-1P* if it satisfies the following conditions:

a. $ID(N) \subset P$, $OD(N) \subset P$, $BD(N) \subset T$, $|ID(N)| = 1$, $|OD(N)| = 1$ and $|BD(N)| = 1$.

Suppose $ID(N) = \{p\}$, $OD(N) = \{p_o\}$ and $BD(N) = \{t\}$.

b. $M(p) = 0$.

c. $\forall t' \neq t$, $W(p, t') = 0$.

d. If $W(p, t) \neq W(t, p_o)$, then $\exists k \in \{1, 2, ...\}$: $kW(p, t) = W(t, p_o)$ or $\exists k' \in \{1, 2, ...\}$: $W(p, t) = k'W(t, p_o)$. And $\forall t' \in {}^{\bullet}p$, $\exists k'' \in \{0, 1, 2, ...\}$: $W(t', p) = k'k''$.

Condition c means that t' is the only output of the input door place p of N.

Replacement R-1P (replacing a generalized reducible subnet-1P with a place $p^{\bullet}$)

Let N be a generalized reducible subnet-1P of PTS. If PTS is transformed to PTS' by replacing N with a place $p^{\bullet}$, then PTS is live (resp., bounded) iff PTS' is live (resp., bounded), provided that the following conditions are satisfied:

a. $P' = P - \{ p, p_o \} + \{ p^{\bullet} \}$.

b. $T' = T - \{t\}$.

c. $$M'(p') = \begin{cases} M(p_o) & \text{if } p' = p^{\bullet} \\ M(p') & \text{if } p' \neq p^{\bullet} \end{cases}.$$

d. $\forall t' \in T - \{t\}$,

$$W'(p', t') = \begin{cases} k'' & \text{if } k'' \neq 0,\ p' = p^{\bullet} \text{ and } t' \in {}^{\bullet}p \\ W(p_o, t') & \text{if } k'' = 0 \text{ and } p' = p^{\bullet} \\ W(p', t') & \text{if } k'' = 0 \text{ and } p' \neq p^{\bullet} \end{cases},$$

$$W'(p', t') = \begin{cases} \frac{k}{k'} W(t', p) + W(t', p_o) & \text{if } p' = p^{\bullet} \\ W(t', p') & \text{if } p' \neq p^{\bullet} \end{cases}.$$

□

This transformation reduces a sequence of place-transition-place to a single place.

Theorem 2.2

Replacement R-1P is correct.

Proof: See [LEE87]. □

Example 2.2 (Figure 2.2) [LEE87]

In PTS, N = <{p, p_o}, {t}, {(p, t), (t, p_o)}> is a generalized reducible subnet-1P such that $W(p,t)/W(t,p_o)=k'$, $W(t_1,p)=k'k_1$ and $W(t_2,p)=k'k_2$. According to Replacement R-1P, PTS can be transformed to PTS' by replacing N with place p^* such that, the following conditions are satisfied in PTS': $W'(t_1,p^*)=k_1$, $W'(t_2,p^*)=k_2$, $W'(t_3,p^*)=W(t_3,p_o)$ and $W'(p^*,t_4) = W(p_o,t_4)$. As a result, since PTS' is live (resp., bounded), PTS is also live (resp., bounded).

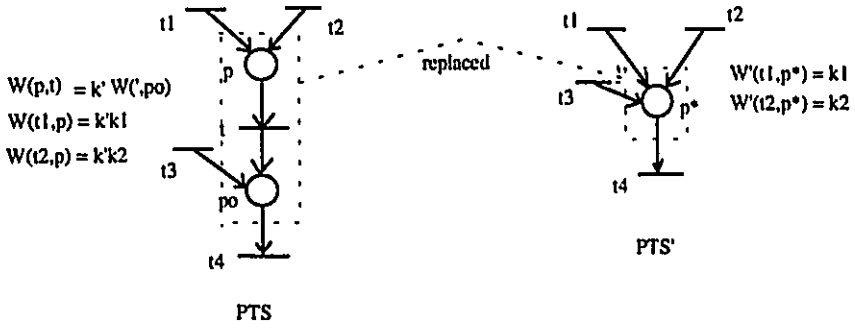


Figure 2.2. The sequence $p-t-p_o$ is replaced with a single place p^*.

2.2 BERTHELOT'S TRANSFORMATIONS [BER84, BER87]

The transformations presented in [BER84, BER87] either eliminate redundant places or fuse some special subsets of places (resp., transitions) into a single place (resp., place). They preserve such properties as liveness and boundedness. These transformations and the conditions for them to preserve these properties are quite simple. However, it is difficult to detect such specific places and transitions since it is based on the entire set of reachable markings of the system.

There are more than ten transformations presented in [BER84, BER87]. Similar to last sect, only two of these transformations will be described to illustrate the main theme of these methods.

One of the transformations in [BER87] eliminates some special places, called *redundant places*, and their connected arcs. A redundant place is a place that contains, for every reachable marking, enough tokens for firing any transitions connected to it. Its formal definition is given below.

Definition 2.5 (redundant place)

A place $p \in S$ is said to be *redundant* w.r.t. $P \subset S$, where $p \in P$, if there exists a "weight function" Φ: $P \cup \{p\} \rightarrow \{1, 2, ...\}$ such that the following conditions are satisfied:

a. $\forall M \in [M_o\rangle$: $\Phi(p)M(p) \geq \sum_{q \in P} \Phi(q)M(q)$

b. $\forall t \in T$: $\Phi(p)M(p, t) \leq \sum_{q \in P} \Phi(q)M(q, t)$

Elimination RP (eliminating a redundant place)

Let p be a redundant place in PTS. If PTS is transformed to PTS' by eliminating p, the tokens in p and the arcs connected to p, then PTS is live (resp., bounded) iff PTS' is live (resp., bounded). □

Theorem 2.3

Elimination RP is correct.

Proof: See [BER87]. □

Example 2.3 (Figure 2.3) [BER87]

Place p_4 is redundant w.r.t. $P = \varnothing$. Place p_5 is redundant w.r.t. $P = \{p_2, p_3\}$ with $\Phi(p_2) = \Phi(p_3) = \Phi(p_5) = 1$. By applying Elimination RP, place $\{p_4, p_5\}$ of PTS, their tokens and connected arcs can be eliminated. As a result, since PTS' is live (resp., bounded), PTS is also live (resp., bounded).

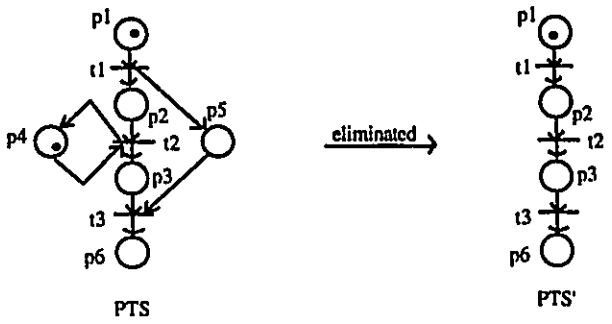


Figure 2.3. Two redundant places, their tokens and connecting arcs are eliminated.

Another transformation proposed in [BER87] is the fusing of *doubled places*. The definition of the doubled places, the fusion and the conditions for it to preserve liveness and boundedness are described below.

Definition 2.6 (doubled places)

Two places p', p ∈ S are said to be *doubled* iff they satisfy the following conditions:

a. $\forall t \in T: |{}^\bullet t \cap \{p', p\}| \leq 1$.

b. If ${}^\bullet t = \{p'\}$, then $\exists t' \in T: {}^\bullet t' = \{p\}$ and $W(t, p) = W(t', p), \forall p \in S$; or, if ${}^\bullet t = \{p\}$, then $\exists t' \in T: {}^\bullet t' = \{p'\}$ and $W(t, p) = W(t', p), \forall p \in S$.

c. $\forall M \in [M_o>, \forall t \in p'^\bullet, \forall s \in S - \{p'\}$ and $M(s) \geq W(s, t): M(p) = 0$. $\forall M \in [M_o>$, $\forall t \in p^\bullet, \forall s \in S - \{p\}$ and $M(s) \geq W(s, t): M(p') = 0$

Condition a indicates that no transition has both p' and p as its input places.

Fusion DP (fusing doubled places)

Suppose that p_1 and p_2 are doubled places in PTS. If PTS is transformed to PTS' by fusing doubled p_1 and p_2 into p_{12} and letting the number of tokens in p_{12} equal to the sum

of the number of tokens in p_1 and in p_2, then PTS is live (resp., bounded) iff PTS' is live (resp., bounded). □

Theorem 2.4

Fusion DP is correct.

Proof: See [LEE87]. □

Example 2.4 (Figure 2.4) [BER87]

In Figure 2.4, places p_3 and p_4 in PTS are doubled since they satisfy the Conditions of Definition 2.6. By Fusion DP, places $\{p_3, p_4\}$ of PTS can be fused into p_{34}. As a result, since PTS' is live (resp., bounded), PTS is also live (resp., bounded).

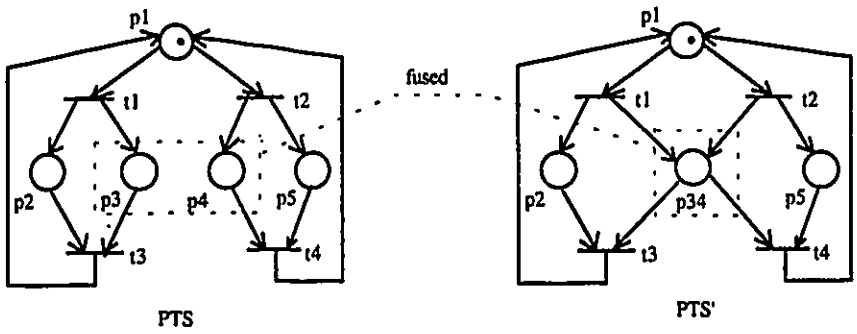


Figure 2.4. Fusing two doubled places into a single place.

2.3 RAMAMOORTHY'S TRANSFORMATIONS [RAM86]

A general transformation presented in [RAM86] preserves not only such desirable properties as liveness and boundedness, but also such undesirable properties as deadness, unboundedness. This transformation replaces a subnet, called *formal well-behaved module* (FWBM), with a transition. Similar to redundant places and doubled places, it is also difficult to detect a since it is based on the entire set of reachability markings of the system.

Definition 2.7 (module, test module and well-behaved module)

A *module* is a subnet N = <S', T', F', W'> of PTS such that $\forall p \in S', \forall t \notin T': p \notin {}^{\bullet}t \cup t^{\bullet}$. A *test module* is a module N = <S', T', F', W'> such that $T' = T_1' \cup T_2'$, $T_1' \cap T_2' = \varnothing$, $T_1' = {}^{\bullet}S'$ and $T_2' = S'^{\bullet}$.A *well-behaved module* (WAM) is a live and bounded test module.

Definition 2.8 (one run)

A WBM is said to be executed *one run* w.r.t. to a marking M(${}^{\bullet}$S') if a sequence of transitions is fired as far as possible until no more transition in OD(WBM) can be fired and a reverse sequence of transitions is fired as far as possible until no more transition in ID(WBM) can be fired.

Definition 2.9 (formal WBM)

A *formal WBM* (FWBM) is a WBM satisfying the following two criterions after executing one run.

1. For all possible M(${}^{\bullet}$S'), the markings at ID(WBM) and OD(WBM) after one run must remain unchanged.
2. WBM is bounded and the markings of the places in WBM remain unchanged after one run. WBM and $t^{\bullet}$ share identical patterns of consuming tokens after WBM is replaced by $t^{\bullet}$.

Replacement FWBM (replacing a FWBM with a transition)

Suppose N is a FWBM of PTS. If PTS is transformed to PTS' by replacing N with a transition $t^{\bullet}$ such that ${}^{\bullet}t^{\bullet}$ = ID(FWBM), $t^{\bullet\bullet}$ = OD(FWBM) and the rest of the system remains unchanged, then PTS is live (resp., dead, bounded and unbounded) iff PTS' is live (resp., dead, bounded and unbounded). □

Theorem 2.5

Replacement FWBM is correct.

Proof: See [RAM86]. □

Example 2.5 (Figure 2.5)

In PTS, a subnet N is a FWBM since it satisfies the conditions given in Definition 2.9. Hence, PTS can be transformed to PTS' by applying Reduction FWBM such that N is replaced with a transition t^*. As a result, since PTS' is live (resp., bounded), PTS is also live (resp., bounded).

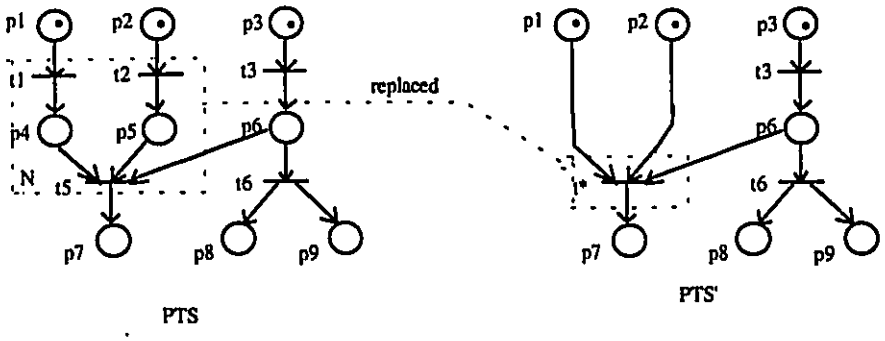


Figure 2.5. A FWBM is replaced with a single transition t^*.

2.4 OTHER TRANSFORMATIONS

Except those presented in the previous sections, many transformations have been studied in the literature. Some of the important ones are quoted below:

a. Equivalence-preserving transformations

A set of exhibited-behavior-equivalence-preserving transformations based on 1-safe system, which is a special PTS, have been presented in [CIN85]. The concept of EB-equivalence has been shown by its author to be the basis for organizational abstraction, which allows us to design distributed systems by refining it. Some

other equivalence-preserving transformations can be found in [GEN90, LUO92, SIM92, WAN91].

b. Reachability tree reduction

A nonprogress-state-preserving transformation has been presented [GOU83] for the communication systems modeled as two finite state machines that communicate by exchanging messages over two one-directional FIFO channels. It splits the task of generating all reachable states into two independent subtasks. In each subtask, only the states reachable by allowing maximal progress for one machine are generated. Some property-preserving reachability tree reductions can be found in [ITO83, WES86, ZHA86, ZHU87].

c. Behavior-preserving refinements

Transformations that replace a place p (resp., transition t) of a contact-free PTS with a net N have been presented in [REI92]. They preserve the behavior that the numbers of tokens flowing in and flowing out of net N are equal to the number of tokens flowing in and flowing out of place p (resp., transition t) after the refinement, assuming that the environment of N is the same as that of place p (resp., transition t). Some other behavior-preserving refinements can be found in [FRE87, SUZ83, VAL79, VOG86].

In summary, most of the existing transformations (except those in [REI92]) are quite straightforward. They either eliminate a special place and its connected arcs or replace a special subnet with a single place or a single transition. All these techniques focus on how to find these special places or special subnets.

Chapter 3

INSERTION AND ELIMINATION

This chapter presents two general classes of transformations on PTSs: *Insertion-I,* which adds a set of places, transitions and arcs into a PTS; and *Elimination-E,* which deletes a set of places, transitions and arcs from a PTS. Since Elimination-E is the reverse process of Insertion-I, our description for the rest of this chapter focuses mainly on Insertion-I. The redundant-places-elimination presented in [BER84, BER87] is a special case of Elimination-E.

3.1 INSERTION-I AND ITS MATRIX REPRESENTATION WITH VERTICAL VECTORS

Insertion-I

A set of places, transitions and arcs are inserted into the PTS.

We shall present two matrix representations of Insertion-I, one in a vertical format (Figure 3.1) and another in a horizontal format (Figure 3.4). They will be used to describe the conditions for preserving invariants.

Matrix representation of Insertion-I with vertical vectors (Figure 3.1):

$$\underbrace{\begin{array}{c} \\ \text{UP} \\ \text{AP} \end{array} \begin{array}{c} \text{UT} \qquad\qquad \text{AT} \\ \begin{pmatrix} X_1 & \cdots & X_i & X_{i+1} & \cdots & X_n \\ Y_1 & \cdots & Y_i & Y_{i+1} & \cdots & Y_n \end{pmatrix} \end{array}}_{\text{V of PTS}} \rightarrow \underbrace{\begin{array}{c} \\ \text{UP} \\ \text{AP} \\ \text{IP} \end{array} \begin{array}{c} \text{UT} \qquad\qquad \text{AT} \qquad\qquad \text{IT} \\ \begin{pmatrix} X_1 & \cdots & X_i & X_{i+1} & \cdots & X_n & 0_{n+1} & \cdots & 0_{n+m} \\ Y_1 & \cdots & Y_i & Y'_{i+1} & \cdots & Y'_n & Y_{n+1} & \cdots & Y_{n+m} \\ 0_1 & \cdots & 0_i & Z_{i+1} & \cdots & Z_n & Z_{n+1} & \cdots & Z_{n+m} \end{pmatrix} \end{array}}_{\text{V' of PTS'}}$$

UP/UT, AP/AT, IP/IT: unaffected, affected, inserted places/transitions.

Figure 3.1. Representation of Insertion-I in vertical format.

Explanation:

a. V is the incidence matrix of PTS and V' is the incidence matrix of PTS'. All 0_k, X_k, Y_k, Y_k' and Z_k are vertical vectors.

b. UP/UT (*unaffected places/transitions*) is the set of places/transitions not connected to any of the inserted transitions/places and AP/AT (*affected places/transitions*) is the set of places/transitions whose attached arcs (or their weights) have been modified.

c. In V', the inserted places IP are represented by the lowest |IP| rows: (0_1 ... 0_i Z_{i+1} ... Z_n Z_{n+1} ... $Z_{n+|IT|}$), where each 0_k is a zero |IP|-vector for k = 1, ..., i. The inserted transitions IT are represented by the rightmost |IT| columns, where each 0_k is a zero |UP|-vector for k = n + 1, ..., n + |IT|.

d. The change of Y_k to Y_k' indicates that some arcs have been inserted between the original places and transitions.

3.2 CONDITIONS FOR INSERTION-I TO PRESERVE PLACE-INVARIANTS

The conditions for Insertion-I to preserve place-invariants are presented below:

Theorem 3.1

Suppose PTS is transformed to PTS' by Insertion-I (Figure 3.1) and U, A and I are subsets of UP, AP and IP, respectively. Then, U ∪ A is a place-invariant of PTS iff U ∪ A ∪ I is a place-invariant of PTS', provided that the following conditions are satisfied:

1. The net flow of every affected transition w.r.t. A ∪ I in PTS' is equal to its net flow w.r.t. A in PTS (i.e., $\forall t_k \in AT$, $NF(\{t_k\}, A \cup I) = NF(\{t_k\}, A)$), or, equivalently,

$$\{Y_k'\}_A + \{Z_k\}_I = \{Y_k\}_A, \quad k = i + 1, ..., n. \tag{3.1}$$

2. The net flow of every inserted transition w.r.t. A ∪ I is zero (i.e., $\forall t_k \in IT$, $NF(\{t_k\}, A \cup I) = 0$), or, equivalently,

$$\{Y_k\}_A + \{Z_k\}_I = 0 \qquad k = n+1, ..., n+|IT|. \qquad (3.2)$$

Proof: We have to show that $\Gamma = (\alpha\ \beta)$ is a place-invariant of PTS iff $\Gamma' = (\alpha\ \beta\ \gamma)$ is a place-invariant of PTS', where α is a 0/1 $|UP|$-vector representing the places of UP such that $\alpha(p) = 1$ iff $p \in U$, β is a 0/1 $|AP|$-vector representing the places of AP such that $\beta(p) = 1$ iff $p \in A$ and γ is a 0/1 $|IP|$-vector representing the places of IP such that $\gamma(p) = 1$ iff $p \in I$.

From Figure 3.1, it is obvious that, for the unaffected columns, we have

$$(\alpha\ \beta\ \gamma)\begin{pmatrix} X_k \\ Y_k \\ 0_k \end{pmatrix} = (\alpha\ \beta)\begin{pmatrix} X_k \\ Y_k \end{pmatrix} \qquad k = 1, 2, ..., i.$$

for the affected columns, we have (by (3.1))

$$(\alpha\ \beta\ \gamma)\begin{pmatrix} X_k \\ Y_k' \\ Z_k \end{pmatrix} = \alpha X_k + (\beta\ \gamma)\begin{pmatrix} Y_k' \\ Z_k \end{pmatrix} = (\alpha\ \beta)\begin{pmatrix} X_k \\ Y_k \end{pmatrix} \qquad k = i+1, ..., n.$$

Also, for the inserted columns, we have (by (3.2)), $(\alpha\ \beta\ \gamma)\begin{pmatrix} 0_k \\ Y_k \\ Z_k \end{pmatrix} = 0$, for $k = n+1, 2, ...,$ $n+|IT|$. Hence, $\Gamma'V' = 0$ iff $\Gamma V = 0$. □

Corollary 3.2

Suppose PTS is transformed to PTS' by Insertion-I (Figure 3.1) and INV is a place-invariant of PTS. Let $U = UP \cap INV$ and $A = AP \cap INV$. If there exists a (possibly empty) subset of inserted places $I \subseteq IP$ such that U, A and I satisfy Conditions (3.1) and (3.2) of Theorem 3.1, then $INV \cup I$ is a place-invariant of PTS'.

Proof: A special case of Theorem 3.1. □

Discussion:

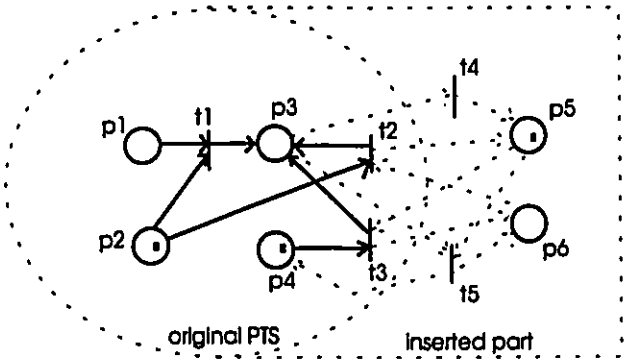
In practice, it is more convenient to apply Corollary 3.2 than Theorem 3.1 for checking the preservation of invariants. With respect to an Insertion-I, the place-invariants of PTS can be separated into two classes. Class 1 contains those which satisfy Corollary 3.2 and are thus preserved (with possible extension) under Insertion-I. Class 2 contains those for which no subset I can be formed to be satisfying Conditions (3.1) and (3.2). They may or may not be preserved under Insertion-I.

Example 3.1 (Figure 3.2)

The set of places IP = $\{p_5, p_6\}$, transitions IT = $\{t_4, t_5\}$ and arcs $\{(t_4\ p_5), (p_5\ t_5), (t_5\ p_6), (p_3\ t_4), (t_2\ p_5), (p_5\ t_3), (p_6\ t_2), (t_3\ p_6), (p_3\ t_5), (t_5\ p_4)\}$ are inserted into PTS. The matrix representation of this insertion is shown below:

$$\begin{array}{c} \\ p1 \\ p2 \\ p3 \\ p4 \end{array}\!\!\begin{array}{c} \begin{array}{ccc} t1 & t2 & t3 \end{array} \\ \begin{pmatrix} -1 & 0 & 0 \\ -1 & -1 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & -1 \end{pmatrix} \end{array} \rightarrow \begin{array}{c} \\ p1 \\ p2 \\ p3 \\ p4 \\ p5 \\ p6 \end{array}\!\!\begin{array}{c} \begin{array}{ccccc} t1 & t2 & t3 & t4 & t5 \end{array} \\ \begin{pmatrix} -1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 \\ 1 & 1 & 1 & -1 & -1 \\ 0 & 0 & -1 & 0 & 1 \\ 0 & 1 & -1 & 1 & -1 \\ 0 & -1 & 1 & 0 & 1 \end{pmatrix} \end{array}$$

V of PTS V ' of PTS'



UP = {p1, p2}, AP = {p3, p4}, IP = {p5, p6}, UT = {t1}, AT = {t2, t3} and IT = {t4,

Figure 3.2. A set of places, transitions and (dotted) arcs are inserted into a PTS.

We have: $A = AP = \{p_3, p_4\}$, $I = IP = \{p_5, p_6\}$ and $INV = \{p_2, p_3, p_4\}$ is a place-invariant of PTS. For each of the affected transitions t_2 and t_3, the net flow w.r.t. $A \cup I = \{p_3, p_4, p_5, p_6\}$ in PTS' is the same as the net flow w.r.t. $A = \{p_3, p_4\}$ in PTS (i.e., Condition (3.1) is satisfied) and both inserted transitions t_4 and t_5 have zero net flow w.r.t. $A \cup I = \{p_3, p_4, p_5, p_6\}$ (i.e., Condition (3.2) is satisfied). It follows from Corollary 3.2 that $INV \cup I = \{p_2, p_3, p_4, p_5, p_6\}$ is a place-invariant of PTS'.

3.3 MATRIX REPRESENTATION OF INSERTION-I WITH HORIZONTAL VECTORS

Matrix representation of Insertion-I with horizontal vectors (Figure 3.3):

$$\begin{array}{c} \\ UP \\ \\ \\ AP \end{array}\!\!\begin{array}{c} \begin{array}{cc} UT & AT \end{array} \\ \begin{pmatrix} B_1 & C_1 \\ \cdots & \cdots \\ B_g & C_g \\ B_{g+1} & C_{g+1} \\ \cdots & \cdots \\ B_m & C_m \end{pmatrix} \end{array} \rightarrow \begin{array}{c} \\ UP \\ \\ \\ AP \\ \\ IP \end{array}\!\!\begin{array}{c} \begin{array}{ccc} UT & AT & IT \end{array} \\ \begin{pmatrix} B_1 & C_1 & 0_1 \\ \cdots & \cdots & \cdots \\ B_g & C_g & 0_g \\ B_{g+1} & C'_{g+1} & D_{g+1} \\ \cdots & \cdots & \cdots \\ B_m & C'_m & D_m \\ 0_{m+1} & C_{m+1} & D_{m+1} \\ \cdots & \cdots & \cdots \\ 0_{m+|IP|} & C_{m+|IP|} & D_{m+|IP|} \end{pmatrix} \end{array}$$

V of PTS V' of PTS '

UP/UT, AP/AT, IP/IT: unaffected, affected, inserted places/transitions.

Figure 3.3. Insertion-I in horizontal format.

In Figure 3.3, all 0_k, B_k, C_k, C'_k and D_k are horizontal vectors. Explanation of the other parts of V and V' is similar to Figure 3.1.

3.4 CONDITIONS FOR INSERTION-I TO PRESERVE TRANSITION-INVARIANTS

Theorem 3.3

Suppose PTS is transformed to PTS' by Insertion-I (Figure 3.3) and U, A and I are subset of UT, AT and IT, respectively. then $U \cup A$ is a transition-vector of PTS iff $U \cup A \cup I$ is a transition-vector of PTS', provided that the following conditions are satisfied,

1. The net flow of every affected place w.r.t. $A \cup I$ in PTS' is equal to its net flow w.r.t. A in PTS (i.e., $\forall p_k \in AP$, $NF(\{p_k\}, A \cup I) = NF(\{p_k\}, A)$), or, equivalently

$$\{C'_k\}_A + \{D_k\}_I = \{C_k\}_A \qquad k = g+1, \ldots, m. \qquad (3.3)$$

2. Every inserted place has zero net flow w.r.t. $A \cup I$ (i.e., $\forall p_k \in AP$, $NF(\{p_k\}, A \cup I) = 0$), or, equivalently,

$$\{C_k\}_A + \{D_k\}_I = 0 \qquad k = m+1, \ldots, m+|IP|. \qquad (3.4)$$

Proof: We have to show that $\Gamma = (\alpha\ \beta)$ is a transition-vector of PTS iff $\Gamma' = (\alpha\ \beta\ \gamma)$ is a transition-vector of PTS', where α is a 0/1 $|UT|$-vector representing the transitions of UT such that $\alpha(t) = 1$ iff $t \in U$, β is a 0/1 $|AT|$-vector representing the transitions of AT such that $\beta(t) = 1$ iff $t \in A$ and γ is a 0/1 $|IT|$-vector representing the transitions of IT such that $\gamma(t) = 1$ iff $t \in I$. It is obvious that, for $k = 1, 2, \ldots, g$,

$$(B_k\ C_k\ 0_k)\Gamma' = B_k\alpha + C_k\beta = (B_k\ C_k)\Gamma.$$

By (3.3), we have, for $k = g+1, \ldots, m$,

$$(B_k\ C'_k\ D_k)\Gamma' = B_k\alpha + (C'_k\ D_k)(\beta\ \gamma) = B_k\alpha + C_k\beta = (B_k\ C_k)\Gamma$$

By (3.4), we have, for $k = m+1, \ldots, m+|IP|$,

$$(0\ C_k\ D_k)\Gamma' = 0.$$

Hence, $V'\Gamma' = 0$ iff $V\Gamma = 0$. □

Chapter 4

ELIMINATING ISOLATABLE PLACES

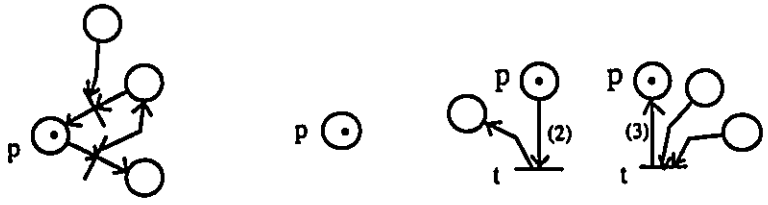
The general Elimination-E studied in Chapter 3 can be regarded as the reverse of Insertion-I. It can also be described in terms of Figure 3.1 or Figure 3.3 by reversing the roles of the two PTSs. The conditions for it to preserve invariants are the same as those given in Theorem 3.1 and 3.2.

In this chapter, we consider a special class of eliminations which satisfy Conditions (3.1) and (3.2). Based on this transformation, an algorithm for finding place-invariants is derived.

4.1 ISOLATABLE, ISOLATED AND NON-ISOLATABLE PLACES

Definition 4.1 (isolatable, isolated, and non-isolatable place)

A place $p \in S$ is said to be *isolatable* iff ${}^\bullet p \cup p^\bullet \neq \emptyset$, $\forall t \in {}^\bullet p$: $\sum_{p' \in {}^\bullet t} W(p', t) \geq W(t, p)$, and $\forall t \in p^\bullet$: $W(p, t) \leq \sum_{p' \in t^\bullet} W(t, p')$. A place $p \in S$ is said to be *isolated* iff ${}^\bullet p = p^\bullet = \emptyset$. A place $p \in S$ is said to be *non-isolatable* if it is neither isolatable nor isolated.



isolatable place isolated place non-isolatable place

Figure 4.1. Isolatable place, isolated place and non-isolatable place.

Lemma 4.1

A place-invariant cannot contain any non-isolatable place.

Proof: A non-isolatable place p is the last case shown in Figure 4.1. If p belongs to a set of places, say P, firing any $t \in p^{\bullet}$ (or ${}^{\bullet}p$) will decrease (or increase) the total number of tokens in P even if the entire $t^{\bullet}$ (or ${}^{\bullet}t$) is in P. Hence, P cannot be a place-invariant. □

4.2 ELIMINATION EIP

Elimination EIP (Eliminating an isolatable place)

Given an isolatable place p of a PTS, Elimination EIP eliminates p and produces a set of affected places AP by the following steps:

1. Let AP = ∅.
2. For every $t \in {}^{\bullet}p$, do the following: Select a subset $\{(p_i,t)\}_{i=1}^{k}$ from the set of incoming arcs of t and, for i = 1, ..., k, decrease a value $W'(p_i,t)$ from the weight $W(p_i,t)$, where $0 < W'(p_i,t) \leq W(p_i, t)$, in such a way that $\sum_{i=1}^{k} W'(p_i,t) = W(t, p)$. Then, let $AP = AP \cup \{p_1, p_2, ..., p_k\}$ and $W(t, p) = 0$.
3. For every $t \in p^{\bullet}$, do the following: Select a subset $\{(t,p_i)\}_{i=1}^{j}$ from the set of the outgoing arcs of t and, for i = 1, ..., j, decrease a value $W'(t,p_i)$ from the weight $W(t,p_i)$, where $0 < W'(t,p_i) \leq W(t, p_i)$, in such a way that $W(p, t) = \sum_{i=1}^{j} W'(t,p_i)$. Then, let $AP = AP \cup \{p_1, p_2, ..., p_j\}$ and $W(p, t) = 0$.
4. Eliminate place p, arcs whose weights have been reduced to zero in Steps 2 and 3 (including all the incoming and outgoing arcs of p) and transitions the weights of all whose attached arcs have been reduced to zero.□

Discussion:

Depending on the selection of the arcs and weights in Step 2 and Step 3, Elimination EIP may reduce PTS to different PTS's and produce different AP's.

When viewed in the reverse order, Elimination EIP may be considered as a special case of Insertion-I which inserts the isolatable place p and a set of transitions, and inserts or modifies an associated set of arcs. In fact, IP consists of row p, IT consists of those columns reduced to zero-vectors, AP consists of all modified rows except row p, AT consists of all modified columns except those reduced to zero, UP consists of unmodified rows and UT consists of unmodified columns.

Theorem 4.2.

In Elimination EIP, let PTS_1 be transformed to PTS_2, p be the isolatable place eliminated and AP be the set of affected places produced. Suppose INV is a place-invariant of PTS_2.

Case 1. If $INV \cap AP = \emptyset$, then INV is also a place-invariant of PTS_1.

Case 2. If $AP \subseteq INV$, then $INV \cup \{p\}$ is a place-invariant of PTS_1.

Proof: In Case 1, since $INV \cap AP = \emptyset$ implies that $INV \subseteq UP$ (i.e., INV is unaffected), it is obvious that INV is also a place-invariant of PTS_1.

In Case 2, in order to apply Theorem 3.1, Elimination EIP is viewed as the inverse of an insertion into PTS_2, resulting in PTS_1. Let $U = INV \cap UP$, $A = INV \cap AP = AP$, $I = IP = \{p\}$, AT be the set of transitions selected in Step 2 and 3 of Elimination EIP, PTS = PTS_2 and PTS' = PTS_1. Since p is isolatable, it follows from the way by which the weights of the arcs are reduced in Elimination EIP that the netflow of every affected transition remains unchanged. That is U, A and I satisfy Conditions (3.1) and (3.2) of Theorem 3.1. Since INV is a place-invariant of PTS_2, $INV \cup \{p\}$ is a place-invariant of PTS_1. □

4.3 ALGORITHM FPI

In general, Theorem 3.1 and 3.2 determine whether a transformation preserves a given invariant or not but do not tell us how to find the transformation and the invariant. The algorithm described below finds a place-invariant iteratively. The main idea is to apply Elimination EIP and Theorem 4.2 in such a way that, in each iteration, if a place-invariant for a complex PTS cannot be found easily, find one for a reduced system produced by Elimination EIP. The algorithm can be described through a tree structure, with its root assigned to the given PTS. In general, each node controls one application of Elimination EIP. Suppose it is unable to find a place-invariant of PTS_{k-1} associated with a node, generate a child of this node by eliminating an isolatable place from PTS_{k-1} and then try to find a special place-invariant of this child. This place-invariant, with the eliminated places along a path from this node to the root, may be used to form a place-invariant of the system. Detail of the algorithm is given below.

Algorithm FPI (to find a place-invariant of a PTS) (Figure 4.2)

Step 1. Let PTS_o = PTS and k = 0.

Step 2. Obtain the set of isolatable places ISP_k of PTS_k (Note 1).

Step 3. • If $ISP_k \neq \emptyset$, then obtain p_k form it. Let $ISP_k = ISP_k - \{p_k\}$ and apply Elimination EIP to eliminate p_k from PTS_k. Set k = k + 1. Let $SPTS_k$ be the collection of reduced systems and SAP_k be the collection of sets of affected places (Note 1). Go to Step 4.

• If $ISP_k = \emptyset$, then one of the following two cases will occur:

Case 1 (k = 0). No place-invariant for PTS_o has been found. Terminate the algorithm (Note 2).

Case 2 (k > 0). Go to Step 4.

Step 4. • If $SPTS_k \neq \emptyset$, then get PTS_k from $SPTS_k$ and AP_k from SAP_k. Let $SPTS_k = SPTS_k - \{PTS_k\}$ and $SAP_k = SAP_k - \{AP_k\}$. *Try* to find a

place-invariant INV_k of PTS_k such that $INV_k \supseteq \bigcup_{i=1}^{k} AP_i - \bigcup_{i=1}^{k-1} \{p_i\}$ (i.e., find a place-invariant containing all the sets of affected places along the path from the root to the current node except the eliminated ones) (Note 4).

One of the following cases will occur:

Case 1 (It is shown that PTS_k has no such place-invariants). Repeat Step 4.

Case 2 (Such a place-invariant INV_k is found). $INV_k \cup \bigcup_{i=0}^{k-1} \{p_i\}$ is a place-invariant of PTS_o. Terminate the algorithm (Note 3).

Case 3 (Existence of such place-invariants cannot be easily found). Go to Step 2.

- If $SPTS_k = \varnothing$, then set $k = k - 1$ and go to Step 3. □

Notes:

1. In general, for each p_k, its associated arcs may be eliminated in different ways. Hence, more than one reduced system and one set of affected places may be obtained. The collections are defined here just for the convenience in describing the algorithm. In practice, we do not have to generate the entire collection of reduced systems in advance but to compute each of them when needed.
2. In this case, it only states that no place-invariant of PTS has been found by the algorithm. It does not necessarily mean that the PTS has no place-invariant.
3. In this case, one place-invariant of PTS has been found by the algorithm. However, this may not be the only place-invariant of PTS.
4. The requirement $INV_k \supseteq \bigcup_{i=1}^{k} AP_i - \bigcup_{i=1}^{k-1} \{p_i\}$ is quite restrictive. Occasionally, some INV_k' of PTS_k without including $\bigcup_{i=1}^{k} AP_i - \bigcup_{i=1}^{k-1} \{p_i\}$ may turn out to be a

place-invariant of PTS_0. But, in general, as shown by Example 4.1, there is no such guaranty.

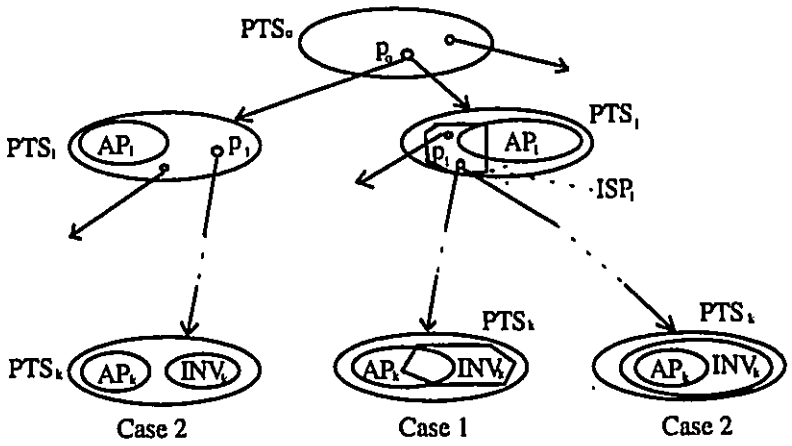


Figure 4.2. Conceptual tree structure of Algorithm FPI.

Theorem 4.3.

If Case 2 of Step 4 is reached at the kth iteration of Algorithm FPI, then for j = k-1, k-2, ..., 0, $INV_k \cup \bigcup_{i=j}^{k-1} \{p_i\}$ is a place-invariant of PTS_j.

Proof: Mathematical induction over the indexes: j = k-1, k-2, ..., 0, will be used. By Theorem 4.2, since $INV_k \supseteq AP_k$, $INV_k \cup \{p_{k-1}\}$ is a place-invariant of PTS_{k-1}. Suppose $INV_k \cup \bigcup_{i=j}^{k-1} \{p_i\}$ is a place-invariant of PTS_j. Since

$$INV_k \cup \bigcup_{i=j}^{k-1} \{p_i\} \supseteq (\bigcup_{i=j}^{k} AP_i - \bigcup_{i=j}^{k-1} \{p_i\}) \cup \bigcup_{i=j}^{k-1} \{p_i\} = \bigcup_{i=j}^{k} AP_i \supseteq AP_j,$$

by Theorem 4.2, $(INV_k \cup \bigcup_{i=j}^{k-1} \{p_i\}) \cup \{p_{j-1}\}$ (i.e., $INV_k \cup \bigcup_{i=j-1}^{k-1} \{p_i\}$) is a place-invariant of PTS_{j-1}. □

4.4 AN EXAMPLE

Example 4.1 (Figure 4.3)

Place p_4 is isolatable in PTS_0. After eliminating p_4 by Elimination EIP, we obtain $AP_1 = \{p_3, p_5, p_6\}$ and PTS_1. Place p_2 is isolatable in PTS_1. After eliminating p_2 by Elimination EIP, we obtain $AP_2 = \{p_3, p_5\}$ and PTS_2. In PTS_2, the place-invariant $INV_2 = \{p_3, p_5, p_6\}$ can be easily found. Since $AP_1 \cup AP_2 - \{p_2, p_4\} = \{p_3, p_5, p_6\} \subseteq INV_2$, $INV_2 \cup \{p_4\} \cup \{p_2\} = \{p_2, p_3, p_4, p_5, p_6\}$ is a place-invariant of PTS_0.

Note that, in PTS_2, $INV_2' = \{p_3, p_5\}$ is also a place-invariant of PTS_2 and $INV_2' \cup \{p_2\}$ is also a place-invariant of PTS_1. But, the latter does not include the affected place $\{p_6\}$ and cannot be a place-invariant of PTS_0.

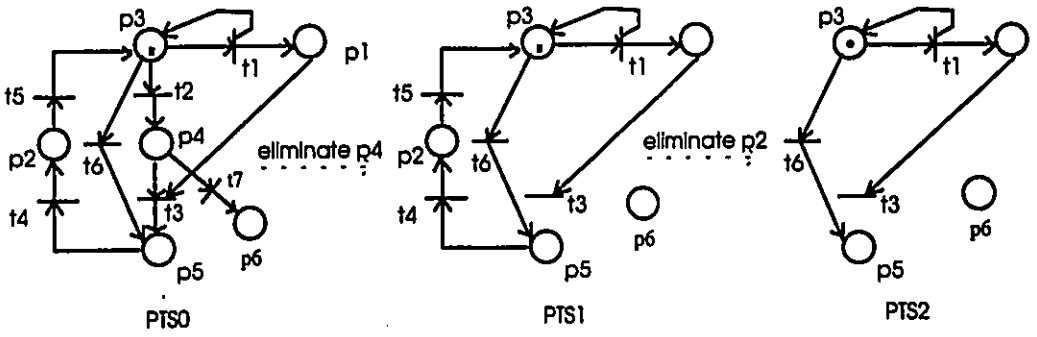


Figure 4.3. To find a place-invariant by eliminating isolated-places.

Chapter 5

REPLACING PLACES OR TRANSITIONS

This chapter presents two other classes of transformations on PTSs: (1) *Replacement RP* (replacement of places) replaces a set of places and arcs with another set of places and arcs. (2) *Replacement RT* (replacement of transitions) replaces a set of transitions and arcs with another set of transitions and arcs.

Every transformation presented in [LEE75, LEE87, RAM86] is a combination of a Replacement RP and a Replacement RT. The double-places-fusing method presented in [BER84, BER87] is a special case of Replacement RP.

5.1 REPLACEMENT RP AND ITS MATRIX REPRESENTATION WITH VERTICAL VECTORS

Replacement RP

A set of places and arcs of PTS is replaced with another set of places and arcs.

Matrix representation of Replacement RP with vertical vectors (Figure 5.1):

Figure 5.1 shows the matrix representations of PTS and PTS'.

$$\begin{matrix} UP \\ AP \end{matrix}\begin{pmatrix} X_1 & \dots & X_n \\ Y_1 & \dots & Y_n \end{pmatrix} \rightarrow \begin{matrix} UP \\ AP' \end{matrix}\begin{pmatrix} X_1 & \dots & X_n \\ Y'_1 & \dots & Y'_n \end{pmatrix}$$

V of PTS V' of PTS'

Figure 5.1. Representation for Replacement RP in vertical format.

Explanation:

a. V is the incidence matrix of PTS and V' is the incidence matrix of PTS'.

b. UP (unaffected places) is the set of places not affected by the replacement and AP is the set of places replaced with those of AP'.

Example 5.1 (Figure 5.2)

The set of places AP = $\{p_2, p_3, p_4\}$ and arcs $\{(p_2\ t_1), (p_3\ t_3), (p_4\ t_4), (t_1\ p_2), (p_2\ t_2), (t_2\ p_3), (t_3\ p_4)\}$ are replaced with the set of places AP' = $\{p_2', p_3'\}$ and arcs $\{(p_2'\ t_1), (p_2'\ t_3), (p_3'\ t_4), (t_1\ p_2'), (t_2\ p_3'), (t_3\ p_3')\}$, respectively.

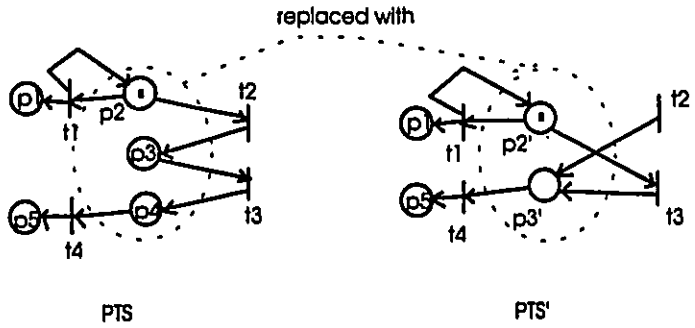


Figure 5.2. Places $\{p_2, p_3, p_4\}$ are replaced with places $\{p_2', p_3'\}$.

As shown in Figure 5.3, Replacement RP can be considered as a combination of an Elimination-E and an Insertion-I.

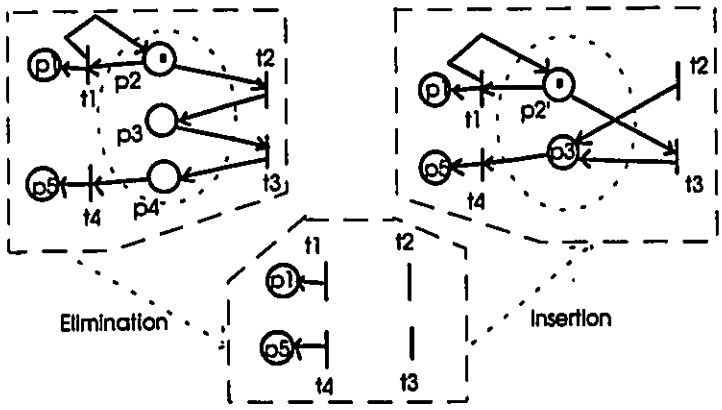


Figure 5.3. A Replacement RP is a combination of an Elimination-E and an Insertion-I.

5.2 CONDITIONS FOR REPLACEMENT RP TO PRESERVE INVARIANTS

The conditions for Replacement RP to preserve invariants are stated in Theorem 5.1 and Theorem 5.2.

Theorem 5.1

Suppose PTS is transformed to PTS' by Replacement RP (Figure 5.1) and U, A and A' are subsets of UP, AP and AP', respectively. Then $U \cup A$ is a place-invariant of PTS iff $U \cup A'$ is a place-invariant of PTS', provided that the net flow of every transition w.r.t. A and A' remains unchanged,. This is

$$\{Y_k\}_A = \{Y'_k\}_{A'}, \qquad k = 1, 2, ..., n. \tag{5.1}$$

Proof: We have to show that $\Gamma = (\alpha\ \beta)$ is a place invariant of PTS iff $\Gamma' = (\alpha\ \beta')$ is a place invariant of PTS', where α is a 0/1 |UP|-vector representing places of UP such that $\alpha(p) = 1$ iff $p \in U$, β is a 0/1 |AP|-vector representing places of AP such that $\beta(p) = 1$ iff $p \in A$ and β' is a 0/1 |AP'|-vector representing places of AP' such that $\beta'(p) = 1$ iff $p \in A'$.

From Figure 5.1 and (5.1), we have, for k = 1, 2, ..., n,

$$(\alpha\ \beta)\begin{pmatrix} X_k \\ Y_k \end{pmatrix} = \alpha X_k + \beta Y_k = \alpha X_k + \beta' Y'_k = (\alpha\ \beta')\begin{pmatrix} X_k \\ Y'_k \end{pmatrix}$$

Hence, $\Gamma V = 0$ iff $\Gamma' V' = 0$. □

Corollary 5.2

Suppose PTS is transformed to PTS' by Replacement RP (Figure 5.1) and INV is a place-invariant of PTS. Let $U = UP \cap INV$ and $A = AP \cap INV$. If there exists a (possibly empty) subset $A' \subseteq AP'$ such that U, A and A' satisfy Condition (5.1) of Theorem 5.1, then $(INV - A) \cup A'$ is a place-invariant of PTS'.

Proof: A special case of Theorem 5.1. □

Discussion:

With respect to a Replacement RP, the place-invariants of PTS can be grouped into two classes. Class 1 contains those which satisfy Corollary 5.2 and are thus preserved under Replacement RP. Class 2 contains those for which no subset A' can be formed to be satisfying (5.1). They may or may not be preserved under Replacement RP.

Example 5.2 (Figure 5.4)

In a PTS represented by V, let subnet AP be replaced with subnet AP', where

$$V = \begin{pmatrix} 0 & 0 & 0 & 1 & -1 & 0 \\ 1 & 0 & -1 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \end{pmatrix}, AP = \begin{pmatrix} 1 & 0 & -1 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \end{pmatrix} \text{ and } AP' = \begin{pmatrix} 1 & -1 & -1 & 0 & 0 & 1 \\ 0 & -1 & -1 & 0 & 1 & 0 \\ 0 & 1 & 1 & -1 & 0 & 0 \end{pmatrix}.$$

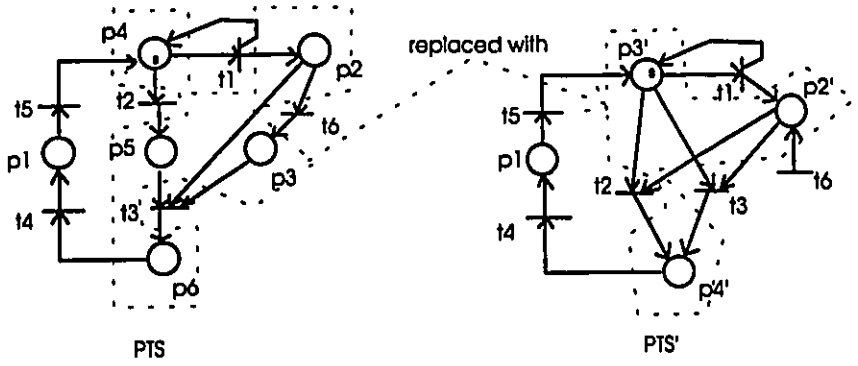


Figure 5.4. Replacing a set of places while preserving a place-invariant.

In this replacement, U = UP = $\{p_1\}$ and AP = $\{p_2, p_3, p_4, p_5, p_6\}$ is replaced with AP' = $\{p'_2, p'_3, p'_4\}$. INV = $\{p_1, p_4, p_5, p_6\}$ is a place-invariant of PTS. Also, there exist A = $\{p_4, p_5, p_6\} \subset$ AP and A' = $\{p'_3, p'_4\} \subset$ AP' such that Condition (5.1) is satisfied. Hence, by Corollary 5.2, (INV - A) $\cup$ A' = $\{p_1, p'_3, p'_4\}$ is a place-invariant of PTS'.

Theorem 5.3

Suppose a set of places AP of PTS is replaced with a single place p' having matrix representation $(y'_1\ y'_2\ \ldots\ y'_n)$ (Figure 5.1) in such a way that the net flow of every transition w.r.t. AP and p' remains unchanged, i.e.,

$$\{ Y_k \}_{AP} = y'_k, \qquad k = 1, 2, \ldots, n. \tag{5.2}$$

Then, a transition-invariant of PTS is also a transition-invariant of PTS'.

Proof: Let Γ be a transition-invariant of PTS. Then $V\Gamma = 0$, implying $(X_1\ X_2\ \ldots\ X_n)\Gamma = 0$ and $(Y_1\ Y_2\ \ldots\ Y_n)\Gamma = 0$. Adding up the last set of equations results in $(\{ Y_1 \}_{AP}\ \{ Y_2 \}_{AP}\ \ldots\ldots\ \{ Y_n \}_{AP})\Gamma = 0$. Hence, by Condition (5.2), $(y'_1\ y'_2\ \ldots\ y'_n)\Gamma = 0$, i.e., Γ is a transition-invariant of PTS'. □

Example 5.3

Suppose the subset of places AP of PTS is merged to a single place p', where

$$V = \begin{pmatrix} -1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix},\ AP = \begin{pmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix} \text{ and } p' = (1\ \ 1\ \ -1\ \ -1).$$

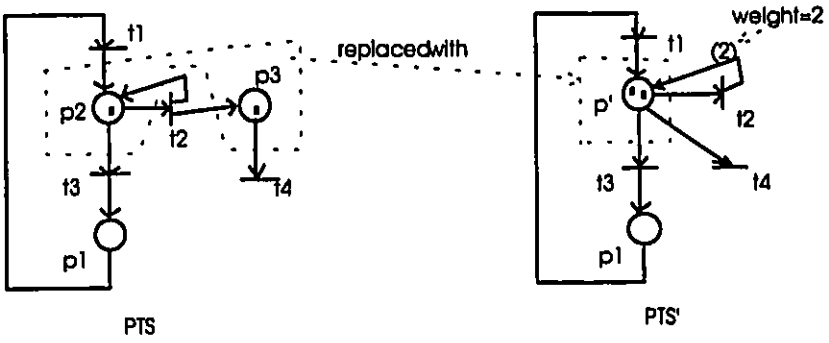


Figure 5.5. Merging a subset of places while preserving transition-invariants.

Since N and p' satisfy Condition (5.2) and (0, 1, 0, 1) is a transition-vector of PTS, it follows from Theorem 5.2 that (0, 1, 0, 1) is a transition-vector of PTS'. That is, $\{t_2, t_4\}$ is a transition-invariant of both PTS and PTS'.

5.3 REPLACEMENT RT AND ITS MATRIX REPRESENTATION WITH HORIZONTAL VECTORS

Replacement RT

A set of transitions and arcs of PTS are replaced with another set of transitions and arcs.

Matrix representation of Replacement RT with horizontal vectors (Figure 5.6)**:**

Figure 5.6 shows the matrix representations of PTS and PTS'.

$$\begin{matrix} \text{UT} & \text{AT} \end{matrix} \quad \begin{matrix} \text{UT} & \text{AT'} \end{matrix}$$

$$\begin{pmatrix} B_1 & C_1 \\ \cdots & \cdots \\ B_m & C_m \end{pmatrix} \rightarrow \begin{pmatrix} B_1 & C_1' \\ \cdots & \cdots \\ B_m & C_m' \end{pmatrix}$$

V of PTS V' of PTS'

Figure 5.6. Representation of Replacement RT in horizontal vectors.

Explanation:

a. V is the incidence matrix of PTS and V' is the incidence matrix of PTS'. All B_k, C_k, B_k' and C_k' are horizontal vectors.

b. UT (unaffected transitions) is the set of unaffected transitions; AT is the set of transitions to be replaced with the set of transitions AT'.

As Replacement RP, Replacement RT can also be considered as a combination of an Elimination-E and an Insertion-I.

5.4 CONDITIONS FOR REPLACEMENT RT TO PRESERVE INVARIANTS

The conditions for Replacement RT to preserve invariants are stated in Theorem 5.4 and Theorem 5.5.

Theorem 5.4

Suppose PTS is transformed to PTS' by Replacement RT (Figure 5.6). If there exist subsets U, A and A' of UT, AT and AT', respectively, such that the net flow of every transition w.r.t. A and A' remains unchanged, then $U \cup A$ is a transition-invariant of PTS iff $U \cup A'$ is a transition-invariant of PTS'. In notation, this condition is

$$\{C_k\}_A = \{C'_k\}_{A'}, \qquad k = 1, 2, ..., m. \tag{5.3}$$

Proof: We have to show that $\Gamma = (\alpha\ \beta)$ is a transition-invariant of PTS iff $\Gamma' = (\alpha\ \beta')$ is a transition-invariant of PTS', where α is a 0/1 |UT|-vector representing transitions of UT such that $\alpha(p) = 1$ iff $p \in U$, β is a 0/1 |AT|-vector representing transitions of AT such that $\beta(p) = 1$ iff $p \in A$, and β' is a 0/1 |AT'|-vector representing transitions of AT' such that $\beta'(p) = 1$ iff $p \in A'$.

From Figure 5.6 and (5.3), it is obvious that

$$(B_k\ \ C_k)(\alpha\ \beta) = B_k\alpha + C_k\beta = (B_k\ \ C'_k)(\alpha\ \beta'), \qquad k = 1, 2, ..., m.$$

Hence, $V\Gamma = 0$ iff $V'\Gamma' = 0$. □

Corollary 5.5

Suppose PTS is transformed to PTS' by Replacement RT (Figure 5.6) and INV is a transition-invariant of PTS. Let $U = UT \cap INV$ and $A = AT \cap INV$. If there exists a (possibly empty) subset of replaced transitions $A' \subseteq AT'$ such that U, A and A' satisfy the Condition (5.3) of Theorem 5.4, then $(INV - A) \cup A'$ is a transition-invariant of PTS'.

Proof: A special case of Theorem 5.4. □

Example 5.4

Suppose the set of transitions (columns) AT of PTS is replaced with the set of transitions (columns) AT', where

$$V = \begin{pmatrix} 1 & -1 & 0 & 1 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{pmatrix}, \quad AT = \begin{pmatrix} -1 & 0 & 1 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \end{pmatrix} \text{ and } AT' = \begin{pmatrix} -1 & 1 \\ 1 & -1 \\ -1 & 1 \end{pmatrix}.$$

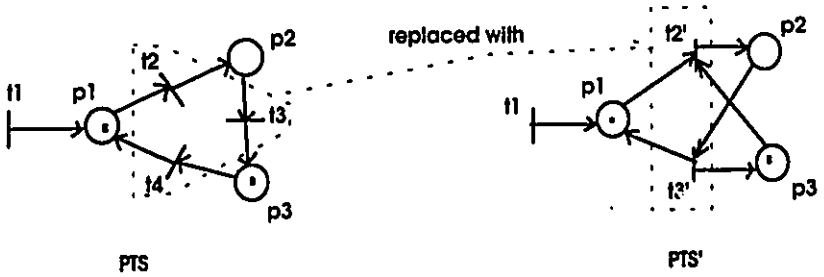


Figure 5.7. Replacing a set of transitions while preserving transition-invariants.

In the Replacement RT, UT = $\{t_1\}$ and AT = $\{t_2, t_3, t_4\}$ is replaced with AT' = $\{t_2', t_3'\}$. Since INV = $\{t_2, t_3, t_4\}$ is a transition-invariant of PTS, there exist U = $\varnothing$, A = AT = $\{t_2, t_3, t_4\}$ and A' = AT' = $\{t_2', t_3'\}$ such that Condition (5.3) is satisfied, it follows from Corollary 5.5 that (INV - A) $\cup$ A' = $\{t_2', t_3'\}$ is a transition-invariant of PTS'.

Theorem 5.6

Suppose that a set of transitions in the subnet $Col(C_1 \ldots C_m)$ of PTS is replaced with a single transition $Col(c_1' \ldots c_m')$ (Figure 5.6), such that the net flow of every place w.r.t. them remains unchanged, i.e.,

$$\{C_k\}_{AT} = c_k', \quad k = 1, 2, \ldots, m. \tag{5.4}$$

Then, a place-invariant of PTS is also a place-invariant of PTS'.

Proof: Let Γ represent a place-invariant of PTS, i.e., $\Gamma V = 0$. From Figure 5.6, we have $\Gamma \bullet Col(B_1 \ldots B_m) = 0$ and $\Gamma \bullet Col(C_1 \ldots C_m) = 0$. By adding up the second set of

equations, we get $\sum_{k=1}^{m} p_k \{C_k\}_{AT} = 0$. It follows from (5.4) that $\sum_{k=1}^{m} p_k c_k' = 0$, i.e., Γ is a place-invariant of PTS'. □

Interpretation of Theorem 5.6:

If the net flow of every place w.r.t. AT is equal to its net flow w.r.t. the merged transition, then a place-invariant of PTS is also a place-invariant of PTS'.

Example 5.5

Suppose the set of transitions (columns) in AT is replaced with transition t', where

$$V = \begin{pmatrix} -1 & -1 & 0 \\ 0 & 1 & -1 \\ 1 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}, \quad AT = \begin{pmatrix} -1 & 0 \\ 1 & -1 \\ 1 & 0 \\ -1 & 1 \end{pmatrix} \quad \text{and} \quad t' = \begin{pmatrix} -1 \\ 0 \\ 1 \\ 0 \end{pmatrix}.$$

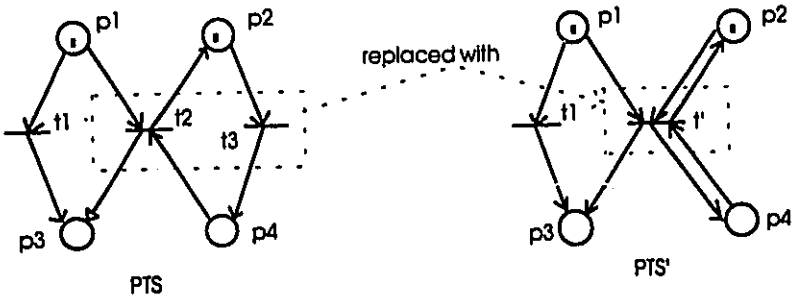


Figure 5.8. Merging a subset of transitions while preserving place-invariants.

Since subnet AT and transition t' satisfy Condition (5.4) and (1, 0, 1, 0) and (0, 1, 0, 1) are place-vectors of PTS, it follows from Theorem 5.6 that (1, 0, 1, 0) and (0, 1, 0, 1) are place-vectors of PTS'. That is, $\{p_1, p_3\}$ and $\{p_2, p_4\}$ are place-invariants of both PTS and PTS'.

Chapter 6

COMPOSITION AND DECOMPOSITION

This chapter presents two more classes of transformations on PTSs: composition and decomposition. A composition connects two PTSs by fusing some of their places and transitions. A decomposition splits a PTS into two subsystems by repeating some of its places or transitions in each of them. Since a composition is the reverse transformation of a decomposition, our description focuses just on compositions for the rest of this section. We consider two special classes of compositions: *Composition CP* connects two PTSs by fusing some places; *Composition CT* connects two PTSs by fusing transitions only. It will be proved that when two PTSs are combined by fusing some of their places, every pair of their place-invariants containing the fused places form a place-invariant for the composite system.

6.1 COMPOSITION CP AND ITS MATRIX REPRESENTATION WITH VERTICAL VECTORS

Composition CP

For two given PTS1 and PTS2, a subset of places AP1 of PTS1 and a subset of places AP2 of PTS2, where |AP1| = |AP2|, are fused together in a one-to-one manner without fusing any of their transitions. (Note: Since no transitions are fused, no arcs are fused also.)

Matrix representation of Composition CP with vertical vectors (Figure 6.1)**:**

Figure 6.1 shows the matrix representation of Composition CP.

$$\begin{matrix} UP1 \\ AP1 \end{matrix}\begin{pmatrix} X_1 & \dots & X_{n1} \\ Y_1 & \dots & Y_{n1} \end{pmatrix} \text{ and } \begin{matrix} AP2 \\ UP2 \end{matrix}\begin{pmatrix} Y'_1 & \dots & Y'_{n2} \\ X'_1 & \dots & X'_{n2} \end{pmatrix} \rightarrow \begin{matrix} UP1 \\ AP \\ UP2 \end{matrix}\begin{pmatrix} X_1 & \dots & X_{n1} & 0_1 & \dots & 0_{n2} \\ Y_1 & \dots & Y_{n1} & Y'_1 & \dots & Y'_{n2} \\ 0_1 & \dots & 0_{n1} & X'_1 & \dots & X'_{n2} \end{pmatrix}$$

V1 of PTS1 V2 of PTS2 V' of PTS'

Figure 6.1. Connecting two PTSs by fusing some of their places.

Explanation:

a. Affected places AP1 of PTS1 and AP2 of PTS2 are the places to be fused. UP1 and UP2 are the unaffected places of PTS1 and PTS2, respectively.

b. For $k = 1, ..., n_1$, each 0_k is a zero |UP2|-vector; for $k = 1, ..., n_2$, each 0_k is a zero |UP1|-vector.

6.2 CONDITIONS FOR COMPOSITION CP TO PRESERVE INVARIANTS

The conditions for Composition CP to combine invariants are presented in the following theorem.

Theorem 6.1

In Composition CP (Figure 6.1), let $P1 \subseteq UP1$, $P2 \subseteq UP2$ and $P \subseteq AP$ be subsets of places, where $|AP| = |AP1| = |AP2|$.

a. If $P1 \cup P$ is a place-invariant of PTS1 and $P \cup P2$ is a place-invariant of PTS2, then $P1 \cup P \cup P2$ is a place-invariant of PTS'.

b. If T_i is a transition-invariant of PTS_i, $i = 1, 2$, then $Col(T_1, T_2)$ is a transition invariant of PTS'.

Proof: (a) Since $P1 \cup P$ is a place-invariant of PTS1, there exists a vector $(0\ \alpha\ 0\ \beta)$, where $\alpha = (1, ..., 1)$ is a |P1|-vector corresponding to the places of the invariant in UP1 and $\beta = (1, ..., 1)$ is a (|P|)-vector corresponding to the places of the invariant in AP1, such that

$$(0\ \alpha\ 0\ \beta)\begin{pmatrix} X_k \\ Y_k \end{pmatrix} = 0, \quad k = 1, ..., n_1.$$

Similarly, since P ∪ P2 is a place-invariant of PTS2, there exists a vector (0 β 0 γ), where γ = (1, ..., 1) is a |P2|-vector corresponding to the places of the invariant in UP2, such that

$$(0\ \beta\ 0\ \gamma)\begin{pmatrix} Y_k' \\ X_k' \end{pmatrix} = 0, \quad k = 1, ..., n_2.$$

Consider another vector (0 α 0 β 0 γ) for PTS', we have

$$(0\ \alpha\ 0\ \beta\ \ 0\ \gamma)\begin{pmatrix} X_k \\ Y_k \\ 0_k \end{pmatrix} = (0\ \alpha\ 0\ \beta)\begin{pmatrix} X_k \\ Y_k \end{pmatrix} = 0,\ k = 1, ..., n_1,$$

and

$$(0\ \alpha\ 0\ \beta\ \ 0\ \gamma)\begin{pmatrix} 0_k \\ Y_k' \\ X_k' \end{pmatrix} = (0\ \beta\ 0\ \gamma)\begin{pmatrix} Y_k' \\ X_k' \end{pmatrix} = 0,\ k = 1, ..., n_2.$$

Hence, (0 α 0 β 0 γ) is a place-vector of PTS', indicating that P1 ∪ P ∪ P2 is a place-invariant of PTS'.

(b) Similar to that of (a). □

Interpretation of Theorem 6.1:

When two PTSs are connected by fusing some of their places, every pair of their place-invariants containing the fused places (resp., transition-invariants) can also be combined to form a place-invariant (resp., a transition-invariant) of the composite PTS.

Example 6.1

Figure 6.2 shows that PTS1 and PTS2 are connected by fusing place p_3 of PTS1 and place p_1' of PTS2. The two fused places p_3 and p_1' become place p of PTS'.

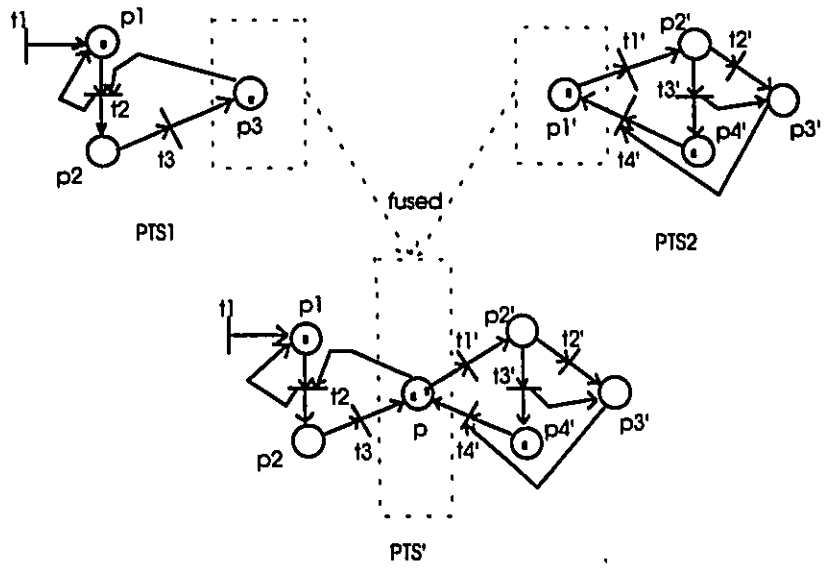


Figure 6.2. PTS1 and PTS2 are connected by fusing one of their places.

The incidence matrices of PTS1, PTS2 and PTS' are given below:

$$V1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \end{pmatrix}, V2 = \begin{pmatrix} -1 & 0 & 0 & 1 \\ 1 & -1 & -1 & 0 \\ 0 & 1 & 1 & -1 \\ 0 & 0 & 1 & -1 \end{pmatrix} \text{ and } V' = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix};$$

$\{p_2, p_3\}$ is a place-invariant of PTS1 and $\{p_1', p_2', p_3'\}$ is a place-invariant of PTS2. By Theorem 6.1, $\{p_2, p, p_2', p_3'\}$ forms a place-invariant of PTS'.

6.3 COMPOSITION CT AND ITS MATRIX REPRESENTATION WITH HORIZONTAL VECTORS

Composition CT

A set of transitions AT1 of PTS1 and a set of transitions AT2 of PTS2, where |AT1| = |AT2|, are fused together in a one-to-one manner without fusing any of their places.

Matrix representation of Composition CT with horizontal vectors (Figure 6.3):

Figure 6.3 shows the matrix representation of Composition CT.

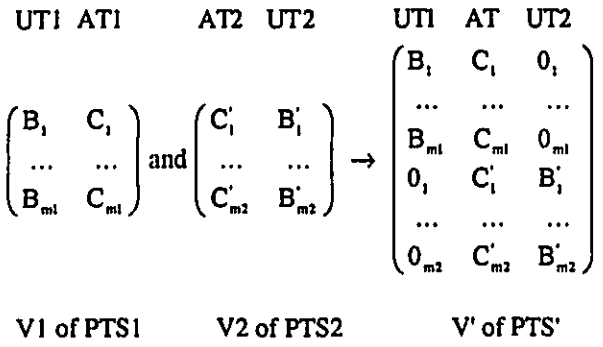


Figure 6.3. Connecting two PTSs by fusing some of their transitions.

Explanation:

Similar to Figure 6.1.

6.4 CONDITIONS FOR COMPOSITION CT TO PRESERVE INVARIANTS

The conditions for Composition CT to combine invariants are presented in the following theorem.

Theorem 6.2

In Composition CT (Figure 6.3), let $T1 \subseteq UT1$, $T2 \subseteq UT2$ and $T \subseteq AT$ be subsets of transitions, where $|AT| = |AT1| = |AT2|$.

a. If (T1, T) is a transition-invariant of PTS1 and (T, T2) is a transition-invariant of PTS2, then (T1, T, T2) is a transition-invariant of PTS'.

b. If P_i is a place-invariant of PTS_i, $i = 1, 2$, then $P_1 \cup P_2$ is a place-invariant of PTS'.

Proof: Similar to Theorem 6.1. □

Chapter 7

FINDING PLACE-INVARIANTS OF THE TRANSPORT PROTOCOL BY REDUCTION

In this chapter, Algorithm FPI (Chapter 4) and Composition CP (Chapter 6) are applied to find the place-invariants of the Transport Protocol.

The Transport Protocol (TP), as proposed by ISO (International Organization for Standardization) and ECMA (European Computer Manufacturers Association), has five different classes that handle errors of increasing severity [ISO8073]. Each class provides services in three phases: Connection Establishment Phase, Data Transfer Phase and Disconnection Phase. The Connection Establishment Phase provides the TP primitives: connect request, connect indication, connect response and connect confirmation for the establishment of an end-to-end transport connection between two transport entities. The Data Transfer Phase serves the exchange of data messages via TP primitives: data request, data indication, expedited-data request and expedited-data indication. The Disconnection Phase uses the TP primitives disconnect request, disconnect indication, disconnect response and disconnect confirmation to terminate a transport connection.

Class 1 provides the ability of error detection and reporting and of recovery from failures. Class 2 provides flow control for merging multiple Transport connections on a single Network connection.

7.1 THE INDIVIDUAL TRANSPORT ENTITIES

Figure 7.1 shows the PTS's of two communicating Transport entities.

The entity initializes itself by transition SI or SI'.

In the idle state (place p_1) and on receipt of a request from the Session Layer, an entity sends (transition RC) a connect request (place CR') and moves to the "waiting for a

connect confirm" state (place p_2). When this confirm message arrives (place CC), the entity passes to the "data transfer" state (place p_4) and can send data (not represented here). Conversely, on receipt of the connect request (place CR'), a remote entity is created (with place p_1' marked); it sends (transition CA') a connect confirm (place CC'), and passes directly to the "data transfer" state.

Once in the data transfer state (places p_4 or p_4'), each entity may end a transport connection by sending (transition RD) a disconnect request (place DR'), and then move to the "waiting for disconnect confirm" state (place p_5). In this state, data messages are discarded. On receipt of a disconnect request, the remote entity returns to the initial state either by sending (transition AD') a disconnect confirm (place DC) if it was in the "data transfer" state, or if is was in waiting for disconnect confirm state (place p_5'). The role of an entity can be switched.

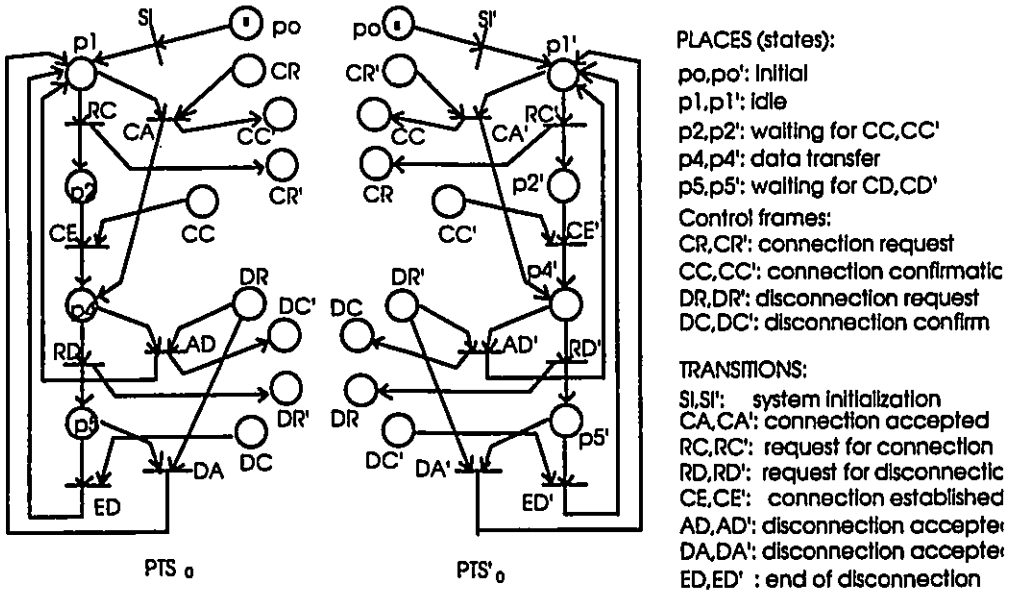


Figure 7.1. Connection and disconnection phases of two Transport entities.

7.2 THE COMMUNICATING TRANSPORT ENTITIES

To show the communication of the two entities, Composition CP (Section 6) is applied to connect PTS_o^* and PTS_o^* by fusing the sets of places $\{p_o, CR, CC', CR', CC, DC', DR', DC\}$ of PTS1 and the set of places $\{p_o', CR, CC', CR', CC, DC', DR', DC\}$ of PTS2 in a one-to-one manner. The resulting PTS is shown in Figure 7.3.

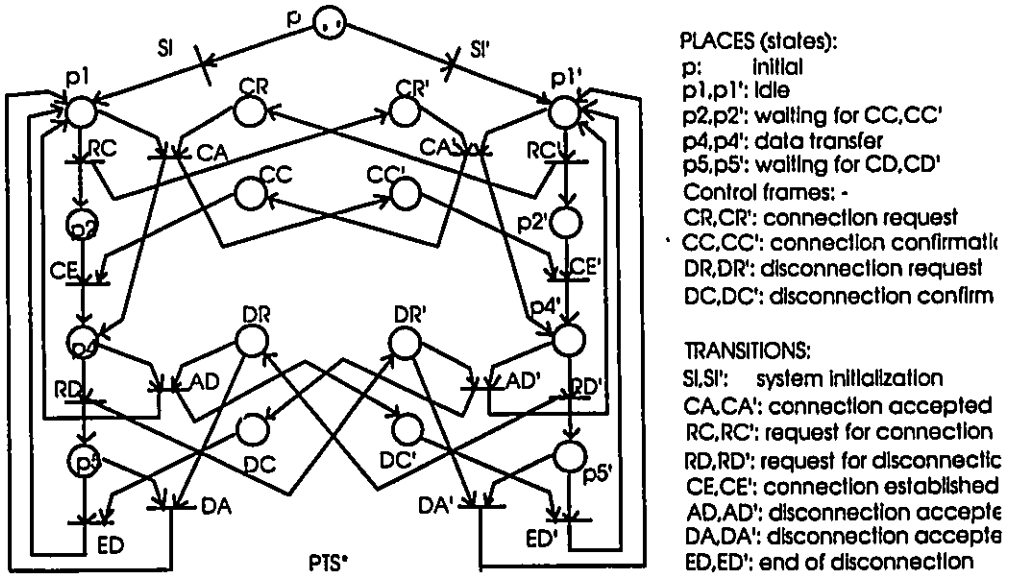


Figure 7.2. Global net model of two communicating Transport entities.

7.3 DETECTION OF PLACE-INVARIANTS

In order to find a place-invariant of the global communicating system, we first find one for each of the entities. Three iterations of Algorithm FPI are applied on PTS_o^* (Table 7.1) and the final PTS_3^* is in Figure 7.3. Then, it is easy to find that $INV_3 = \{p_o, p_1\}$ is a place-invariant of PTS_3^*. Since $AP_3 \cup AP_3 \cup AP_3 - \{p_2, p_4, p_5\} = \{p_1\} \subset INV_3$, by Theorem 4.3, $INV_3 \cup \{p_2, p_4, p_5\} = \{p_o, p_1, p_2, p_4, p_5\}$ is a place-invariant of PTS_o^*.

.iteration	original PTS	isolatable place p	reduced PTS	AP w.r.t. p	place-invariant
1	PTS'_0	p_2	PTS'_1	p_1, p_4	------
2	PTS'_1	p_4	PTS'_2	p_1, p_5	------
3	PTS'_2	p_5	PTS'_3	p_1	p_0, p_1

Table 7.1. Three iterations of Algorithm FPI on PTS'_0.

Similarly, by applying Algorithm FPI on PTS^*_0, $\{p'_0, p'_1, p'_2, p'_4, p'_5\}$ is found to be a place-invariant of PTS^*_0.

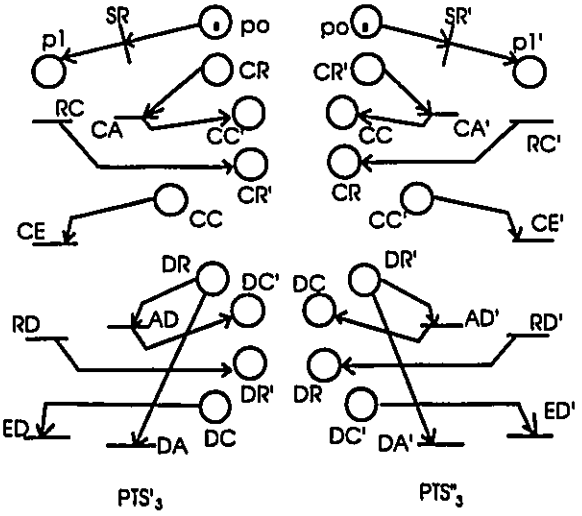


Figure 7.3 Reduced nets of the individual Transport entities.

Let p be the place created by fusing p_0 and p'_0. Since $\{p_0, p_1, p_2, p_4, p_5\}$ and $\{p'_0, p'_1, p'_2, p'_4, p'_5\}$ are the place-invariants of PTS'_0 and PTS^*_0, respectively, it follows from Theorem 6.1 that the set of places $\{p, p_1, p_2, p_4, p_5, p'_1, p'_2, p'_4, p'_5\}$ is a place-invariant of global PTS^*.

Chapter 8

CONCLUSION AND FURTHER STUDIES

Property-preserving transformations are useful techniques for the verification of complex systems. However, at this stage of the arts, the transformations reported in the literature are quite restrictive and simple and hence have a rather limited scope of application. In this thesis, we have considered five general classes of transformations on PTSs, namely, insertion, elimination, replacement, composition and decomposition, and proposed the conditions for them to preserve place-invariants and transition-invariants. We have also proposed a special class of elimination called Elimination-EIP, which eliminates an isolatable place, a set of transitions (possibly empty) and a set of arcs. It leads to the proposal of a constructive algorithm for finding place-invariants of PTSs. Our algorithm derives place-invariants for a system from its reduced systems. This is different from the traditional way which determines place-invariants by obtaining the non-negative integer solutions of a set of linear integer equations derived from the Petri-net.

Two computational aspects of our algorithm require further investigation. The first is about the determination of the place-invariants in each iteration of our algorithm. As it is now, only a place-invariant which contains the places in all sets of affected places AP_k's is accepted. In general, this is quite restrictive and it may not be easy to derive place-invariants for the original system in this way. How to make use of other types of place-invariants in each iteration requires further studies. Secondly, in Step 3, different ways of eliminating an isolatable place p_k may end up with different simplified place/transition systems PTS_k's and sets of affected places AP_k's. Just like reachability analysis, how to explore these PTS_k's and AP_k's to generate the different paths becomes a difficult problem and requires further studies.

Another aspect is worth further investigation. In this thesis, only reservation of invariants is considered. But, many other properties are also important. As mentioned in [BER82, REI85], invariants of PTSs can be used to verify other properties such as liveness, boundedness, etc.. Hence, the transformations proposed in this thesis, though mainly for place-invariants and transition-invariants, are also useful for verifying these properties. However, a more direct approach is to find transformations essentially for these properties. This opens many areas for future research.

Though many transformations on place/transition systems have been studied in recent years [BER84, BER87, CIN85, LEE85, LEE87, RAM86, REI92], there is no systematic study of them. A detailed survey on these problems is being prepared by Professor T. Y. Cheung and this author.

REFERENCES:

[APT85] Krzysztof R. Apt, Logics and Models of Concurrent Systems, Springer-Verlag, 1985.

[BER88] P. Berlinguette and D. Gueraichi, "The alternating bit protocol in LOTOS: 'textual' and 'graphical' representation", Technical Report TR-88-25, Dept. of Computer Science, Univ. of Ottawa, 1988.

[BER82] G. Berthelot and R. Terrat, "Petri net theory for the correctness of protocols", IEEE Trans. on Communications, Vol. COM-30, No. 12 (Dec. 1982), pp. 2497-2505.

[BER84] G. Berthelot and G. Lri-Iie, "Reductions of nets and parallel programs", Lecture Notes in Computer Science, Vol. 222 (1984), pp. 19-40.

[BER87] G. Berthelot, "Transformations and decompositions of nets", Lecture Notes in Computer Science, Vol. 254 (1987), pp. 359-376.

[BIL88] J. Billington, G. Wheeler and M. Wilbur-ham, "PROTEAN: A high-level Petri Net tool for the specification and verification of communication protocol", IEEE Trans. on Software Engineering, Vol. SE-14, No. 3 (1988), pp. 301-316.

[BOC87] B.v. Bochmann, "Usage of protocol development tools: the results of a survey", Protocol Specification VII, edited by H. Rudin and C.H. West (1987), pp. 139-164.

[BRA90] F. Brady, A. Boshier, D. Pitt and B. Szczygiel, "One2one - a tool for translating ASN.1 to ACT one", Formal Description Techniques, III - FORTE'90, edited by J. Quemada, J. Manas and E. Vazquez (1990), pp. 539-542.

[BRA83] D. Brand and P. Zafiropulo, "On communicating finite-state machines", J. of ACM, Vol. 30, No. 2 (Apr. 1983), pp. 323-342.

[BRA87] W. Brauer, W. Reisig and G. Rozenberg (ed.), Petri Nets: Central Model and Their Properties, Springer-Verlag, 1987.

[BRO83] S.D. Brookes and W.C. Rounds, "Behavioural equivalence relations induced by programming logic", ICALP 83, Lecture Notes in Computer Science, Vol. 154 (1983), pp. 97-108.

[BUD87] S. Budkowski and P. Dembinski, "An introduction to Estelle: a specification language for distributed systems", Computer Networks and ISDN Systems 14 (1987), pp. 3-23.

[CCI92] Specification and description language, CCITT Z. 100, International Consultative Committee on Telegraphy and Telephony, Geneva, 1992.

[CHE89] M.S. Chen and D.D. Dimitrijevic, "Dynamic state exploration in quantitative protocol analysis", Protocol Specification, Testing, and Verification IX, edited by E. Brinksma, G. Scollo and C.A. Vissers (1989), pp. 327-338.

[CHE88] T.Y. Cheung and Y. Zhu, "A Petri-net-based method for specifying distributed systems and deriving executable graphical LOTOS and ESTELLE", Technical Report, TR-88-04, Dept. of Computer Science, Univ. of Ottawa, 1988.

[CIN85] F.D. Cindio, L. Pomello and C. Simone, "Exhibited-behavior equivalence and organizational abstraction in concurrent system design", Proc. 5th IEEE Intern. Conf. on Distributed Computing Systems, Denver, Colorado (1985), pp. 486-495.

[DAN77] A. Danthine, "Petri nets for protocol modeling and verification", Proc. Computer Networks and Teleprocessing Symp. (Oct. 1977), pp. 663-685.

[DIA87] M. Diaz, "Petri net based models in the specification and verification of protocol", in Petri Nets: Central Model and Their Properties, Springer-Verlag, 1987.

[FRE87] J. Freudenmenn, "Development of communication software by stepwise refinement", Protocol Specification, Testing, and Verification VII, edited by H. Rudin and C.H. West (1987), pp. 391-404.

[GEN90] Hartmenn J. Genrich, "Equivalence transformations of Petri nets", Lecture Notes in Computer Science, Vol. 424, (1990).

[GOU83] M.G. Gouda and Y.T. Yu, "Protocol validation by maximal progress state exploration", IEEE Trans. on Communications, Vol. COM-32, No. 2 (1983), pp. 94-97.

[HAI83] B.T. Hailpern and S.S. Owicki, "Modular verification of computer communication protocol", IEEE Trans. on Communications, Vol. COM-31, No. 1 (1983), pp. 56-68.

[HOA78] C.A.R. Hoare, "Communicating sequential processes", Communications of the ACM, Vol. 21, No. 8, 1978, pp. 666-677.

[HOA81] C.A.R. Hoare, S.D. Brookes and A.W. Roscoe, "A theory of communicating sequential progresses", Technical Report PRG-16, Oxford University Computing Laboratory, Programming Research Group, Oxford, England, 1981.

[ISO89a] "Information processing systems - Open Systems Interconnection - ESTELLE - a formal description technique based on an Extended State Transition Model", ISO/IEC 9074, International Organization for Standardization, Geneva, 1989.

[ISO89b] "Information processing systems - Open Systems Interconnection - LOTOS - a formal description technique based on the temporal ordering of observational behavior", ISO/IEC 8807, International Organization for Standardization, Geneva, 1989.

[ITO83] M. Itoh and H. Ichikawa, "Protocol verification algorithm using reduced reachability analysis", Trans. of the IECE of Japan, Vol. E66, No. 2 (1983), pp.88-93.

[LAM84] S. Lam and U. Shankar, "Protocol verification via projections", IEEE Trans. on Software Engineering, Vol. SE-10, No. 6 (1984), pp. 325-342.

[LAM94] S. S. Lam and A. Udaya Shanker, "A theory of interfaces and modules I-composition theorem", IEEE Trans. on Software Engineering, Vol. SE-20, No. 1 (Jan., 1994), pp. 55-71.

[LAU74] K. Lautenbach and H. Schmid, "Use of Petri nets for proving correctness of concurrent process systems". Information Processing 1974 - North Holland Pub. Co. (1974), pp. 187-191.

[LEE85] H. Lee-Kwang and J. Favrel, "Hierarchical reduction methods for analysis and decompositions", IEEE Trans. on Systems, Man, Cybernetics. Vol. SMC-15 (Mar. 1985), pp. 272-280.

[LEE87] H. Lee-Kwang, J. Favrel and P. Baptiste, "Generalized Petri net reduction methods", IEEE Trans. on Systems, Man, Cybernetics. Vol. SMC-17 (Mar. 1987), pp. 297-303.

[LIN87] F.J. Lin, P.M. Chu and M.T. Liu, "Protocol verification using reachability analysis: The state space explosion problem and relief strategies", ACM SIGCOMM'87 Symp., edited by J.J. Garcia-Luna-Aceves (Aug. 1987), pp. 126-135.

[LUO92] G. Luo, G.v. Bochmann, A. Das and C. Wu, "Failure-equivalent transformations of transition systems to avoid internal actions", Information Processing Letters, Vol. 44, No. 6 (Dec. 1992), pp.333-343.

[MEM86] G. Memmi and J. Vautherin, "Analysis nets by the invariant method", Lecture Notes in Computer Science, Vol. 254 (1986), pp. 300-336.

[MIL80] Robin Milner, A Calculus of Communicating Systems, Lecture Notes in Computer Science, Vol. 92, Springer-Verlag, 1980.

[OBA87] A. Obaid and L. Logrippo, "An atomic calculus of communicating systems", Protocol Specification, Testing and Verification VII, edited by H. Ruding and C.H. West (1987), pp. 91-104.

[PET77] J. Peterson, "Petri nets", ACM Computing Surveys, Vol. 9, No. 3 (1977), pp. 223-252.

[RAM86] C. V. Ramamoorthy and Y. Yaw, "A Petri net reduction algorithm for protocol analysis", SIGCOMM'86 Symp., Communications Architecture & Protocol, Stowe, Vermont (August, 1986), pp. 305-327.

[REI85] W. Reisig, Petri Nets - An Introduction, Springer-Verlag (1985), pp. 77-97.

[REI91] W. Reisig, "Petri nets and algebraic specifications", Theoretical Computer Science, Vol. 80 (1991), pp. 1-34.

[REI92] W. Reisig, A Primer in Petri Net Design, Springer-Verlag, 1992.

[SCH81] R.L. Schwartz and P.M. Melliar-Smith, "Temporal logic specification of distributed systems", Proc. 2nd International Conf. Distributed Computing Systems, Paris (Apr. 1981), pp. 446-454.

[SHA90] S.M. Shatz, P.S. Kajka and A.S. Chauhan, "Formal modeling and automated analysis of the LAPD protocol", Computer Networks and ISDN Systems 18 (1989/90), pp. 293-314.

[SIM92] C. Simone and M.A. Marson, "The application of EB-equivalence rules to the structural reduction of GSPN models", J. of Parallel and Distributed Computing 15 (1992), pp. 296-302.

[SUZ83] I. Suzuki and T. Murata, "Stepwise refinement of transitions and places", J. of Comp. Syst. Science 27 (1983), pp. 51-76.

[VAL79] R. Valette, "Analysis of Petri nets by stepwise refinement", J. of Computer Science, Vol. 18, No. 1 (Feb. 1979), pp. 35-46.

[VOG86] Walter Vogler, "Behavior and life preserving refinements of Petri nets", Institut Fur Informatik TU Munchen Postfach 202420 D-8000 Munchen 2, West Germany.

[WAN91] Guoqiang Wang, "A unified approach for equivalence relations of indeterministic distributed systems (with application to network protocol)", A master thesis, Univ. of Ottawa, Ontario, Canada, March 1991.

[WES86] C.H. West, "Protocol validation by random state exploration", Protocol Specification, Testing, and Verification VI, edited by B., Sarikaya and G.v. Bochmann (1986), pp. 233-242.

[YUA89] M.C. Yuang and A. Kershenbaum, "Parallel protocol verification using the localized approach: the two-phase algorithm", Protocol Specification, Testing, and Verification IX, edited by E. Brinksma, G. Scollo, C.A. Vissers (1989), pp. 339-356.

[ZHA86] Z.R. Zhao and G.v. Bochmann, "Reduced reachability analysis of communication protocols: a new approach", Technical Report, Universite de Montreal, 1986.

[ZHU87] Y. Zhu and T.Y. Cheung, "A new distributed breadth-first-search algorithm", Information Processing Letters, Vol. 25 (1987), pp. 1088-1989.