Reprinted from

INTEGRATION

the VLSI journal

INTEGRATION, the VLSI journal 26 (1998) 211–223

A new approach in feature interaction testing

Masahide Nakamura*, Tohru Kikuno

Department of Informatics and Mathematical Science, Osaka University, 1–3 Machikaneyama, Toyonaka city, Osaka 560-8531, Japan

ELSEVIER

INTEGRATION

the VLSI journal

www.elsevier.com/locate/vlsi

Aims and Scope

INTEGRATION's aim is to act as a professional journal covering every aspect of the VLSI area, with an emphasis on cross-fertilization between various fields of science and the design and verification of VLSI circuits. Individual issues will feature peer-reviewed tutorials and articles, reviews of recent publications, and a calendar of events. The intended coverage of the journal can be assessed by examining the following (nonexclusive) list of topics:

- specification methods and languages
- high-level synthesis for VLSI systems
- algorithms implemented in parallel architectures
- logic synthesis and finite automata, testability
- layout design of VLSI circuits
- testing
- formal verification
- integrated CAD systems and silicon compilers
- systems engineering
- VLSI architectures
- algorithms
- process-technology
- VLSI theories

ELSEVIER INTEGRATION, the VLSI journal 26 (1998) 211–223

INTEGRATION the VLSI journal

A new approach in feature interaction testing

Masahide Nakamura*, Tohru Kikuno

Department of Informatics and Mathematical Science, Osaka University, 1–3 Machikaneyama, Toyonaka city, Osaka 560-8531, Japan

Abstract

Feature Interaction (FI) is known as a kind of inconsistent conflicts between new and existing services on the telecommunication networks. In order to detect and eliminate FI, FI testing is needed which checks whether FI occurs or not for given multiple services. In this paper, we first introduce practical examples of FI and then formulate FI testing problem using finite state machine. Next, we review the conventional testing methods. Since the conventional methods generally utilize exhaustive state enumeration for the testing, thus so-called state explosion problem becomes a bottle-neck for the testing. To avoid this, we propose a new approach by means of P-invariants of Petri net. Theoretically, the P-invariant method provides only necessary condition for FI testing. However, experimental result shows that it essentially works as necessary and sufficient condition for practical services, and that it realizes much more efficient testing than the conventional ones. Therefore, the proposed approach is well applicable to practical FI testing. © 1998 Elsevier Science B.V. All rights reserved.

Keywords: Telecommunication services; Feature Interaction; FI testing; Finite state machine; Petri net

1. Introduction

The recent advancement of VLSI technology accelerates down-sizing and functional enrichment in telecommunication network switching systems. As a result, various new services are being developed and deployed on telecommunication networks in order to accommodate customer's requirements.

When the new services are added to the existing systems, functional conflicts between new and existing services can happen, which may lead the system to inconsistent behaviors, even to be broken down. This is known as a *Feature Interaction (FI) problem* [2,12]. As the number of services increases, the potential number of such FIs becomes combinationally large. Hence, systematic ways for dealing with FI are strongly required in order for the essential solution of the FI problem. FI

* Corresponding author. Tel.: + 81 6 850 6567; fax: + 81 6 850 6569; e-mail: masa-n@ics.es.osaka-u.ac.jp.

PII: S0167-9260(98)00029-7

testing is one of the most important issues to tackle the FI problem. The FI testing in this paper is, for a given pair (X,Y) of services, to check whether the interactions occur or not between X and Y.

The telecommunication services are usually modeled as finite state machines (FSMs), and FIs are defined on the undesirable states (such as deadlock states) in the FSM. Therefore, the straightforward approach to realizing FI testing is to enumerate all possible reachable states by exhaustive search, then identify such undesirable states [4,7,8,11]. This approach is quite simple but powerful in the sense that all interactions are exactly identified based on necessary and sufficient condition. However, the number of states in the FSM exponentially grows in the number of users and features, which is the so-called state explosion problem. Therefore, the application of this approach is limited to relatively simple services with small number of users.

In order to overcome this problem, we propose an alternative method which extensively utilizes the P-invariants of Petri net [9,10]. The P-invariants give us the necessary condition for the states saying that if a state s is reachable from the initial state, then P-invariant equation w.r.t. s must hold. The P-invariants can be calculated without costly state enumeration. We first construct a set of undesirable states at which FI can occur, then check if the states are actually reachable or not by means of P-invariants. Theoretically speaking, although the P-invariants give only necessary condition, the experimental results show that it essentially works as necessary and sufficient condition for practical FI testing. Thus, the proposed approach is applicable to more complex services.

The remainder of the paper is organized as follows: Section 2 introduces practical examples and general concept of FI and scope of this paper. In Section 3, we formalize the FI testing problem by means of a state transition model. Section 4 reviews the conventional FI testing methods. We present a new testing method in Section 5, and perform an experimental evaluation in Section 6. Finally, Section 7 concludes the paper with future works.

2. Feature interaction

2.1. Practical examples

Although there are a lot of instances of FIs reported, here we present typical three examples in practical telecommunication services [13]. In the following, A, B, C and D denote the users in network.

Example 1: CW service and CF service

Call Waiting (CW): This service allows the subscriber to receive an additional call while he is talking. Suppose that x subscribes to CW. Even when x is busy talking with y, x can receive an additional call from third party z.

Call Forwarding (CF): This service allows the subscriber to have his incoming calls forwarded to another number. Suppose that x subscribes to CF and that x specifies z to be a forwarding address. Then, any incoming call to x is automatically forwarded to z.

Interaction CW&CF: Assume that A subscribes to both CW and CF. Suppose that (1) A is talking with B, (2) C is ready to dial, and (3) D is in the A's forwarding address and is idle. Then, if

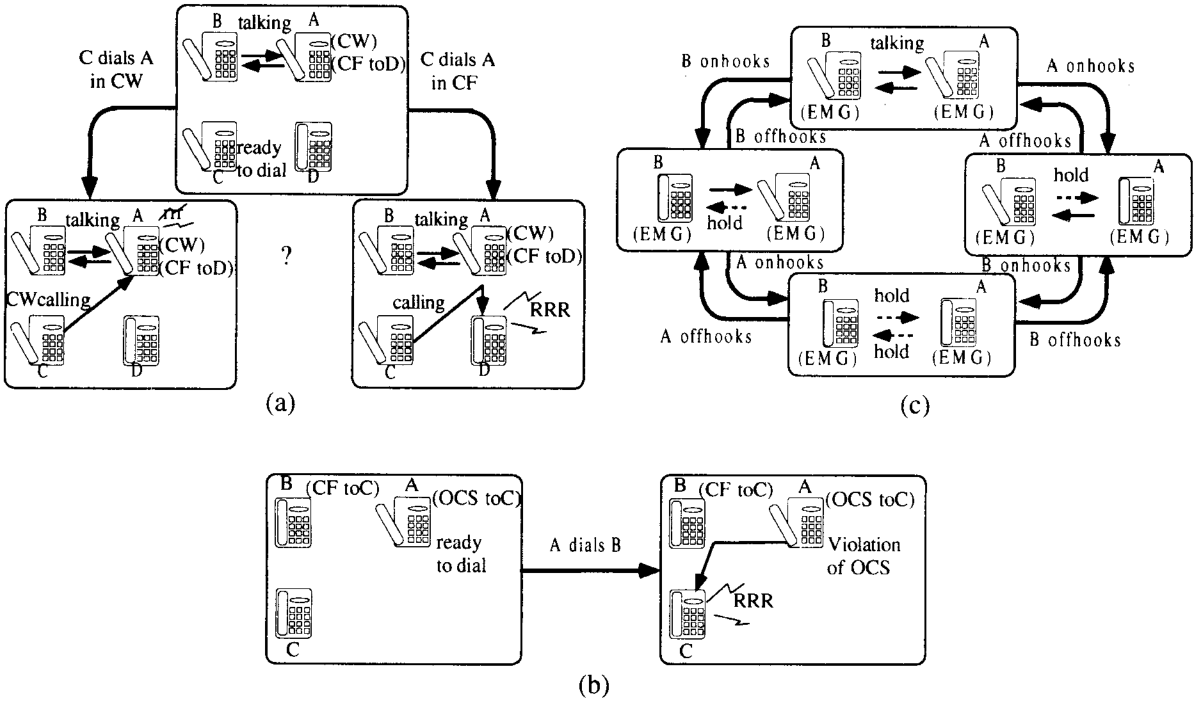


Fig. 1. Examples of FIs.

C dials A, should the call from C to A be received by A's CW feature, or should be forwarded to D by CF feature? This non-determinism will make A confused, thus should be avoided (see Fig. 1a).

Example 2: OCS service and CF service

Originating Call Screening (*OCS*): This service allows the subscriber to specify that outgoing calls be either restricted or allowed according to a screening list. Suppose that x subscribes to OCS and that x puts y in a screening list. Then, any outgoing call to y from x is restricted, while any other call to $z(z \neq y)$ from x is allowed.

Call Forwarding (*CF*): The same as the one in Example 1.

Interaction OCS&CF: Suppose that (1) A is an OCS subscriber and restricts the outgoing call to C, and (2) B is CF subscriber and sets the forwarding address to C. At this time, if A dials B, the call is forwarded to C and A will be calling C. This nullifies A's call restriction to C (see Fig. 1b).

Example 3: EMG services

Emergency call (*EMG*): This service is usually deployed on police station and fire station. In case of the emergency accident, the call will be on hold even when the caller mistakenly onhooks. Suppose that x is a police station on which EMG is deployed, and that y made a call to x and is now busy talking with x. Then, even when y onhooks, the call is on hold without being disconnected. Followed by that, if y offhooks, the held line goes back to connected line and y can talk with x again. In order to disconnect the call, x has to onhook.

Interaction EMG_A *&* EMG_B: Suppose that both A and B subscribe to EMG and are talking to each other. Here, if A onhooks, the call is on hold by B's EMG. At this time, if A offhooks, the call reverts to the talking state. On the other hand, if B onhooks, the call is also held by A's EMG without being disconnected. Symmetrically, this is true when B onhooks first. Thus, neither A nor B can disconnect the call. As a result, the call falls into a loop from which it never returns to the idle state (see Fig. 1c). More instances can be referred to [3,12].

2.2. Concept of feature interaction

The definition of FI is documented as follows:

- "Feature interactions happen when different features of a software system interfere with each other. The feature interaction problem has been most thoroughly documented in telecommunication systems, but it occurs in other systems as well" [5], or
- "Feature interactions are understood to be all interactions that interfere with the desired operation of the features ... " [2].

These definitions are really abstract, however, there is no definition that can clearly explain all FIs, yet. This is because the patterns of the instances are so various that all FIs cannot be dealt in a single way. Therefore, FI problem had been analyzed and resolved in an adhoc manner.

Fifteen years ago, CCITT (presently, ITU-T) proposed the standards for new telecommunication platform called intelligent network (IN) and a set of services (CS-1) [13]. Along with the enrichment of new services, the potential number of FIs combinationally increases. At the same time, considering the telecommunication systems, the problem size becomes very large and complex. Moreover, it is impossible to renew whole system with throwing away the large legacy system in order to accommodate it to incoming services. These facts would be a serious problem which prevents the rapid creation and deployment of new services.

Hence, since then, many researchers and workers have been trying to establish the systematic methods which are alternative to the conventional adhoc techniques. These methods include classification, modeling, detection, resolution, architecture and so on [1,5,6].

2.3. Classification and scope of paper

The first step towards a systematic solution is to classify such instances into several categories [3,7,11]. For example, Ohta et al. [11] defined the following four types of FIs.

(a) *non-determinism*: An event can simultaneously activate two or more functionalities of different services. As a result, it cannot be determined which services should be activated.
(b) *transition to abnormal state*: By combining multiple services, the service execution reaches the abnormal states which have never been defined on each single service.
(c) *loop*: The service execution is trapped into a loop from which the execution never returns to the initial state.
(d) *deadlock*: Functional conflicts of different services cause a mutual prevention of their execution, which results in a deadlock.

Consider again the examples in the previous section. According to this classification, CW&CFV belongs to (a), OCS&CF does to (b), EMG_A&EMG_B does to (c). Although many researchers have proposed their own methods for the solution of the FI problem, there is unfortunately no single method which can cover all types of FIs, yet.

In this paper, we especially focus on the non-deterministic interactions in the above (a), and try to propose the FI testing method for them. That is, for a given pair (X, Y) of services, to check if the non-deterministic FIs occur between X and Y.

3. FI modeling

3.1. Rule-based specification

In this paper, we assume that the service S is given as a set R_S of *rules*, each of which specifies a functionality of the service in terms of a state transition. Since the rule-based service description is quite easy to understand and good in modularity, it is widely studied towards the practical use [7,8,11].

A *rule* consists of *pre-condition, event* and *post-condition.* For example, a rule of CW can be described as

	/* *pre-condition*	*event*	*post-condition* */
*cw*4:	$CW(x), talk(x, y), dialtone(z)$	$[dial(z, x)]$	$CW(x), talk(x, y), CW\text{-}calling(z, x).$

The pre-(post)condition is a list of *predicates*, and the event is a predicate. This rule *cw*4 specifies a functionality of CW, meaning that "Suppose that x is a subscriber of CW, x is talking with y, and z receives a dialtone. At this time, if z dials x, then x can receive an additional call from z". Also, a rule of CF is described as follows:

$$cf10{:}\quad CF(y, z), dialtone(x), idle(z) \qquad [dial(x, y)] \qquad CF(y, z), calling(x, z).$$

which means "Suppose that y is a CF subscriber and put z in its forwarding address, x receives a dialtone, and z is idle. At this time, if x dials y, then x will be calling z".

Next, a state is defined as a list of *instances* of predicates which have constants (representing actual users) as arguments. For example, the following state s means that "user A subscribes to both CW and CF forwarded to D, A is talking with B, C receives dialtone, and D is idle".

$$s = CW(A), CF(A, D), talk(A, B), dialtone(C), idle(D)$$

The *initial state* s_0 can be defined as a state in which all users are idle, for example:

$$s_0 = idle(A), idle(B), idle(C), idle(D)$$

3.2. State transition model

Consider a rule r and a state s. We first instantiate r by applying a *substitution* θ which substitutes all variables in r with some constants. Let $r\theta$ be the resultant instance of rule r based on θ. If all predicates in the pre-condition of $r\theta$ are included in s, then we say r is *enabled* for θ at s. When r is enabled, it can be *applied*. The application of r to s changes the *current state* s to the *next state* s' by replacing the pre-condition of $r\theta$ in s by post-condition of $r\theta$.

For example, consider rule $cf10$ and state s in Section 3.1. First, we apply a substitution $\theta = \{x|C, y|A, z|D\}$ to $cf10$. As a result, we obtain an instance of $cf10$:

$$cf10\theta\text{:}\quad CF(A, D), dialtone(C), idle(D) \qquad [dial(C, A)] \qquad CF(A, D), calling(C, D).$$

Since all predicates in the pre-condition of $cf10\theta$ are included in s, $cf10$ is enabled for θ at s. If we apply $cf10$ to s, the next state s' is obtained by rewriting the pre-condition of $cf10\theta$ with the post-condition as follows.

$$s' = CW(A), CF(A, D), talk(A, B), calling(C, D)$$

This state transition means that if the event dial(C,A) occurs at state s, then C is calling D by a CF functionality ($cf10$).

If there exists a sequence of state $s_0, s_1 \ldots, s_n = s$ such that s_i is the next state of s_{i-1}, then we say s is *reachable* from s_0. The set of all reachable states obtained from the set of rules R and the initial state s_0 is denoted by $REACH(R, s_0)$.

3.3. Definition of non-deterministic FI

For given set R of rules and initial state s_0, a state s is *non-deterministic state* iff

(a) $s \in REACH(R, s_0)$ and

(b) There exists a pair of rules $r_i, r_j \in R$ such that r_i and r_j are, respectively, enabled for certain θ_i and θ_j at s, and that the events of $r_i\theta_i$ and $r_j\theta_j$ are identical.

The pair (r_i, r_j) of rules with respect to the non-deterministic state s is called a *conflicting rules pair*, which is the cause of the non-deterministic interaction.

For given R and s_0, let $U(R, s_0)$ denote the set of all states, and $NDT(R, s_0)$ denote the set of states satisfying the above condition (b), (we may simply represent them as U, NDT, $REACH$ unless confusion). Then, by definition, the set of non-deterministic states is the intersection of $REACH(R, s_0)$ and $NDT(R, s_0)$. Let R_P and R_Q be the given sets of rules of services P and Q, respectively. Then, the **FI testing problem** in this paper is to check if any state s exists with $s \in NDT(R, s_0) \cap REACH(R, s_0)$ for $R = R_P \cup R_Q$ and s_0.

For example, consider again the rules $cf10$, $cw4$ and state s in Section 3.1. Suppose that R_{cw} and R_{cf} are the sets of rules of CW and CF, respectively, and that $cw4 \in R_{cw}$ and $cf10 \in R_{cf}$. Then, the rules $cf10, cw4 \in R_{cw} \cup R_{cf}$ are simultaneously enabled for $\theta = \{x|C, y|A, z|D\}$ and $\theta' = \{x|A, y|B, z|C\}$ at s, and the events of $cf10\theta$ and $cw4\theta'$ are identically dial(C,A). Thus, $s \in NDT(R_{cw} \cup R_{cf}, s_0)$. If $s \in REACH(R_{cw} \cup R_{cf}, s_0)$, then s is a non-deterministic state and ($cf10$, $cw4$) is a conflicting rules pair. Thus, we can see the functional conflict between CW and CF causing the non-deterministic FI. This explains exactly the non-deterministic interaction CW&CF mentioned in Example 1.

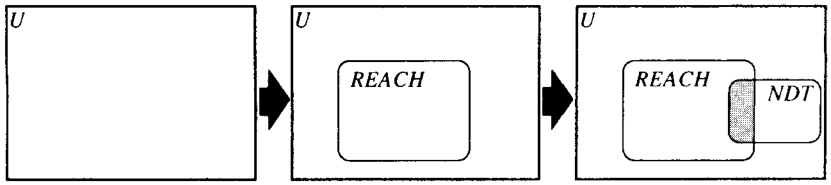


Fig. 2. Outline of conventional method EXH.

4. Conventional FI testing method

The straightforward and conventional approach to the FI testing is to perform exhaustive search of all reachable states. The algorithm, denoted by EXH, is defined as follows. The input of the algorithm is the sets R_P and R_Q of rules for given services P and Q, and initial state s_0.

4.1. FI testing Algorithm EXH:

1. Enumerate all possible reachable states by exhaustive rule applications from s_0 and $R_P \cup R_Q$.
2. For each enumerated state s, check whether s is a non-deterministic state or not. If any enumerated state is not a non-deterministic state, then report "FI is not detected". Otherwise, report "FI is detected".

Fig. 2 depicts the outline of EXH. The following proposition characterizes the algorithm EXH:

Proposition 1. *For given sets of rules R_P and R_Q, and the initial state s_0,*

$$\text{Non-deterministic state exists.} \Leftrightarrow \text{Algorithm EXH reports "FI is detected".}$$

The exhaustive approach is quite simple but powerful. Hence, it is adopted by most of the conventional FI testing methods [4,7,8,11]. However, when the number of rules and the number of users become large, this approach takes a lot of time and space due to the exponential growth of $REACH$, which is the so-called *state explosion problem.* Therefore, the application is limited to the relatively simple services with a few users.

5. New approach

5.1. Converting to Petri net

As defined in Section 3.1, the rule-based service description consists of pre-condition, event and post-condition. This structure is well suited to a Petri net structure consisting of input places,

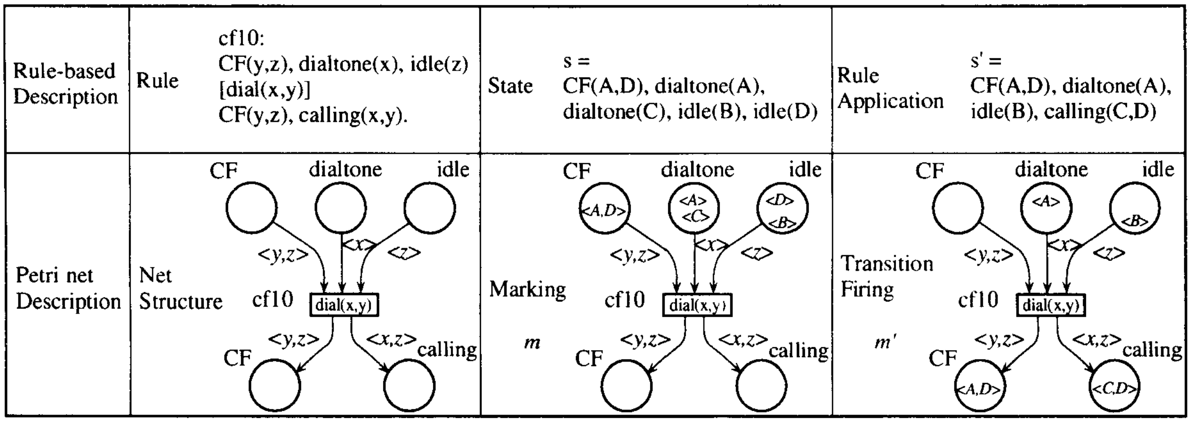


Fig. 3. Correspondence between rule-based description and Petri net.

a transition and output places. This fact motivates us to transform the rule-based description into a Petri net model.

The transformation is performed in the following way:

(a) For each rule r, generate a transition t (depicted by a box), and put the event of r on t.
(b) For each predicate $p(x_1, \ldots, x_k)$ in pre(or post)-condition of r, generate a place p (depicted by a circle) and an arc (depicted by an arrow) from p to t (or t to p, respectively), with labelling $\langle x_1, \ldots, x_k \rangle$.

By doing this, the rules are transformed into a Petri net structure (strictly speaking, a colored Petri net structure). Putting tokens on corresponding places, we can represent each state as a marking. Moreover, a state transition by rule application is exactly isomorphic to the firing of colored Petri net [9]. Fig. 3 depicts the correspondence between rule-based description and Petri net.

For example, consider the marking m in Fig. 3. Let us make transition *cf*10 fire according to the firing rule. First, we substitute all variables appearing on the incident arcs of *cf*10 with some constants. Here, we apply $\{x|C, y|A, z|D\}$. Since each input place of *cf*10 contains enough tokens specified on the corresponding arc, *cf*10 can fire. If *cf*10 fires, then tokens are removed from input places and added to output places in accordance with the substituted labels on arcs. So, we obtain the next marking m' of m. Thus, we can correspond the firing of *cf*10 to the occurrence of dial(C, A).

5.2. P-invariant method

After mapping the rule-based description into Petri net, we can utilize the powerful analysis method of Petri net, called P-invariant method. Intuitively, P-invariant method is to find equations that are satisfied for all reachable markings (i.e., states) of a considered Petri net. The basic idea is that we first assign a *weight* to each place, then make a weighted sum of tokens on all places. If the weights are nicely chosen, the weighted sum of the tokens is equally reserved before and after the transition firing. This forms an *invariant* which always holds on all reachable states. In the colored Petri nets, the weight on each place is specified in terms of linear function. Here, we introduce two kinds of linear functions: (1) *id* s.t. $id\ \langle x \rangle = \langle x \rangle$ and (2) $p12$ s.t. $p12\ \langle x, y \rangle = \langle x \rangle + \langle y \rangle$.

For example, let us consider two markings m and m' in Fig. 3, and assign *id* and $p12$ as weights of places in the following way.

$$\begin{array}{lcccc} & CF & dialtone & idle & calling \\ Y = [& p12, & id, & id, & p12\ ] \end{array}$$

Now, let us calculate the weighted sum of tokens by applying Y to m (denoted by Y^*m).

$$\begin{aligned} Y^*m &= p12\langle A, D\rangle + id\langle A\rangle + id\langle C\rangle + id\langle B\rangle + id\langle D\rangle + p12\phi \\ &= \langle A\rangle + \langle D\rangle + \langle A\rangle + \langle C\rangle + \langle B\rangle + \langle D\rangle = 2\langle A\rangle + \langle B\rangle + \langle C\rangle + 2\langle D\rangle \end{aligned}$$

Similarly, we also apply Y to m' which is the next marking of m.

$$\begin{aligned} Y^*m' &= p12\langle A, D\rangle + id\langle A\rangle + id\langle B\rangle + p12\langle C, D\rangle \\ &= \langle A\rangle + \langle D\rangle + \langle A\rangle + \langle B\rangle + \langle C\rangle + \langle D\rangle = 2\langle A\rangle + \langle B\rangle + \langle C\rangle + 2\langle D\rangle = Y^*m \end{aligned}$$

Thus, before and after transition *cf*10 fires, the weighted sum of tokens is preserved. Such a weight assignment Y of a Petri net N is called P-invariant of N, and can be calculated from net structure of N by algebraic computation [9]. The following is an important theorem which provides a necessary condition of the reachability of states [9].

Theorem 1. *Let Y be P-invariant of a given Petri net N, and let m and m_0 be markings. Then*

$$m \text{ is reachable from } m_0 \Rightarrow Y^*m = Y^*m_0.$$

Thus, for an arbitrary marking (state) s, if $Y^*s \neq Y^*s_0$, then we can conclude s is not reachable from initial state. This P-invariant method can be performed without enumerating all possible states, hence, it does not suffer from the state explosion problem [10]. Instead, even if $Y^*s = Y^*s_0$, we cannot derive any decision on the reachablity of s because the P-invariant gives us only the necessary condition.

5.3. Proposed FI testing method

The outline of the proposed method is shown in Fig. 4. In the figure, the sets $REACH$ and NDT are the same as those in Fig. 2. $P\text{-}INV$ denotes the set of states s satisfying the P-invariant equation $Y^*s = Y^*s_0$, which is a superset of $REACH$. In the proposed method, we first determine the set NDT, then delete unreachable state from NDT by evaluating the P-invariant equation. The algorithm, denoted by PINV, is defined as follows.

FI testing algorithm PINV:

1. From $R_P \cup R_Q$ and state s_0, construct a Petri net N as in Section 5.1. Then, calculate P-invariant(s) Y of N. Define the set NDT of states and initialize it to be empty.

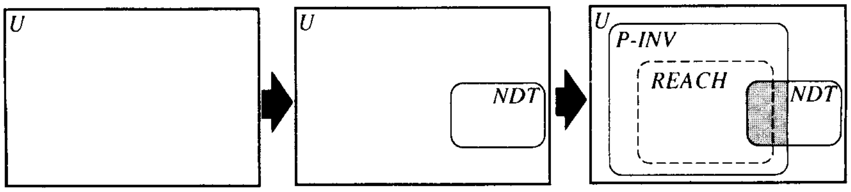


Fig. 4. Outline of proposed method PINV.

2. For each pair of rules r_i, r_j in $R_A \cup R_B$ which have same event symbol,
   - Apply the substitution θ_i and θ_j to r_i and r_j so that the events of $r_i\theta_i$ and $r_j\theta_j$ are identical.
   - Generate a state s by combining the pre-conditions of $r_i\theta_i$ and $r_j\theta_j$.
   - Put s in NDT.
3. For each s in NDT, evaluate the invariant equation. If $Y^*s \neq Y^*s_0$, then delete s from NDT. Otherwise, leave s in NDT.
4. Let the resultant NDT be RES. If $RES = \emptyset$, report "FI is not detected". Otherwise, report "FI is suspected".

The calculation of NDT is easily done by combining pre-conditions of two rules with the same events. For example, consider again $cw4$ and $cf10$ in Section 3.1. First, we apply substitutions $\theta = \{x|A, y|B, z|C\}$ and $\theta' = \{x|C, y|A, z|D\}$ to $cw4$ and $cf10$ in order to make the events identical (dial(C,A)). Then, by combining the pre-conditions of $cw4\theta$ and $cf10\theta'$, we obtain the following state:

$$CW(A), CF(A, D), talk(A, B), dialtone(C), idle(D)$$

This state is the same as state s in Section 3.1, and it belongs to NDT as mentioned in Section 3.4. The following proposition characterizes algorithm PINV. We should note that the reverse of the proposition does not necessarily hold, theoretically speaking.

Proposition 2. *For given sets of rules R_P and R_Q, and the initial state s_0,*

Non-deterministic state s exists. $\Rightarrow$ *Algorithm PINV reports "FI is suspected".*

6. Experimental evaluation

In order to show the effectiveness, we apply the proposed FI testing method to practical services. For the experiment, we have developed a software which can execute both algorithms EXH and PINV. Also, we have prepared the rule-based description for the following five practical services: Call Waiting (CW), Call Forwarding (CF), Direct Connect (DC), Denied Origination (DO), Denied Termination (DT) [13,14]. For each pair out of five services, we try to check if FI occurs or not. We performed two cases of FI testing: one with three users and one with four users.

Table 1
Result of experiment (measured on Sun SS-UA1)

Services (# of rules)	EXH					PINV				
		Three users		Four users			Three user		Four users	
	Testing result	States	Time (s)	States	Time (s)	Testing result	States	Time (s)	States	Time (s)
CW + CF(36)	Detected (5)	102746	4706.4	N/A	N/A	Suspected (5)	423	3.6	616	5.9
CW + DC(27)	None	9592	270.4	693805	66321	None	297	1.8	400	2.9
CW + DT(26)	Detected (1)	7120	187.2	181480	16128	Suspected (1)	285	1.8	388	2.8
CW + DO(26)	None	3480	89.3	71856	6125	None	285	1.7	384	2.7
CF + DC(26)	None	65410	1851.5	N/A	N/A	None	99	0.6	144	1.1
CF + DT(25)	Detected (6)	38584	1006.1	N/A	N/A	Suspected (6)	105	0.7	172	1.1
DC + DO(25)	None	17775	447.6	N/A	N/A	None	75	0.5	108	0.8
DC + DT(16)	None	5390	69.6	296811	9780	None	48	0.3	68	0.3
DC + DO(16)	Detected (2)	4654	57.9	257086	8182	Suspected (2)	57	0.3	80	0.3
DT + DO(15)	None	1450	15.7	19675	497.8	None	33	0.2	44	0.2

Table 1 summarizes the result of the experiment. The column "States" shows the number of states required for each FI testing method. For each of EXH and PINV, we count the states in REACH of EXH and in NDT of PINV, respectively (See Fig. 2 and Fig. 4). The column "Time" shows the elapsed CPU time for each algorithm. The number in column "testing result" is the number of conflicting rules pairs, which cause the non-deterministic FIs at the non-deterministic states.

We can see two facts from the table. First, the state explosion problem was successfully avoided by PINV. The state space and time needed for PINV are significantly smaller than those of EXH. Moreover, with four users, EXH suffered from the state explosion problem and could not complete some cases (represented by N/A) due to the limitation of memory (# of states $>$ 1 000 000). On the other hand, PINV never faced with the state explosion, and was actually able to handle cases with more users.

Second, all FIs suspected by PINV are actually detected by EXH. That is, the necessary condition of P-invariant method essentially worked as necessary and sufficient condition for these practical services. Hence, we can expect PINV to provide the high-quality FI testing for practical services.

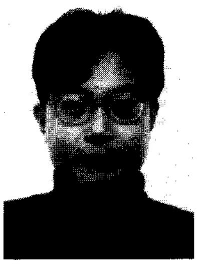
7. Conclusion

In this paper, we have proposed a new FI testing method. By extensively utilizing the P-invariant method, we can successfully avoid the state explosion problem. Also, experimental evaluation has shown that the necessary condition works as the necessary and sufficient condition for practical services. Therefore, we can conclude that the proposed PINV is applicable to more complex services with many users than the conventional EXH. In order to clarify why PINV worked as a necessary and sufficient condition, we have to continue further research on the relationship between P-invariants and some constraints of practical services. The extension of PINV to other types of FIs is also one of our future works.

References

[1] W. Bouma, H. Velthuijsen, (Eds.), Feature Interactions in Telecommunications Systems II, IOS Press, 1992.
[2] E.J. Cameron, H. Velthuijsen, Feature interactions in telecommunications systems, IEEE Commun. Magazine 31 (8) (1993) 18–23.
[3] E.J. Cameron, N.D. Griffeth, Y-J. Lin, M.E. Nilson, W.K. Schnure, H. Velthuijsen, A feature interaction benchmark for IN and Beyond, Proc. 2nd Workshop on Feature Interactions in Telecommunications Systems, IOS Press, 1994, pp. 1–23.
[4] C. Capellmann, P. Combes, J. Pettersson, B. Renard, J.L. Ruiz, Consistent interaction detection – A comprehensive approach integrated with service creation, Proc. 4th Workshop on Feature Interactions in Telecommunications Systems, July 1997, pp. 183–197.
[5] K.E. Cheng, T. Ohta, (Ed.), Feature Interaction in Telecommunications III, IOS Press, 1993.
[6] P. Dini, R. Bautaba, L. Logrippo, (Eds.), Feature Interaction in Telecommunication Networks IV, IOS Press, 1997.
[7] A. Gammelgaard, E.J. Kristensen, Interaction detection, a logical approach, Proc. 2nd Workshop on Feature Interactions in Telecommunications Systems, 1994, pp. 178–196.
[8] Y. Harada, Y. Hirakawa, T. Takenaka, N. Terashima, A conflict detection support method for telecommunication service descriptions, IEICE Trans. Commun. E75-B (10) (1992) pp. 986–997.

[9] K. Jensen, Coloured Petri Nets, EATCS Monographs on Theoretical Computer Science, vol.1–2, Springer, Berlin, 1992.
[10] J.B. Jorgensen, K.H. Mortensen, Modelling and analysis of distributed program execution in BETA using coloured Petri nets, Proc. 17th Int Conf. on Application and Theory of Petri net, June 1996, pp 249–268.
[11] T. Ohta, Y. Harada, Classification, detection and resolution of service interactions in telecommunication services, Proc. 2nd Workshop on Feature Interactions in Telecommunications Systems, 1994, pp. 60–72.
[12] P. Zave, Feature interactions and formal specifications in telecommunications, IEEE Comput. 26 (8) (1993) 20–30.
[13] ITU-T Recommendations Q.1200 Series., Intelligent Network Capability Set 1 (CS1), September 1990.
[14] Bellcore, LSSGR Features Common to Residence and Business Customers I, II, III, Issue 2, July 1987.



Masahide Nakamura received his B.E and M.E degrees in computer science from Osaka University in 1994 and 1996, respectively. He is currently working for his Ph.D. degree in the same university. He is a Research Fellow of the Japan Society for the Promotion of Science (JSPS Research Fellow). He has also received the Paper Award from the Telecommunication Advancement Foundation in 1997.

His research interests include design, verification and testing of telecommunication services and communication protocols. He is a student member of IEEE and IEICE.

Tohru Kikuno was born in 1947. He received his M.Sc. and Ph.D. degrees from Osaka University in 1972 and 1975, respectively. He was at Hiroshima University from 1975 to 1987. He has been a Professor of the Department of Informatics and Mathematical Science, Osaka University, since 1990. His research interests include analysis and design of fault-tolerant systems, quantitative evaluation of software development process and design of testing procedure of communication protocols.

He has been active in the program committee of many international conferences such as FTCS, ISORC, ATS and RTSCA. He is a member of IEICE, IPSJ, IEEE, ACM. He has also received the Paper Award from the Institute of Electronics, Information, and Communication Engineers of Japan in 1993.

Instructions to Authors

Preparation of manuscripts. Manuscripts should be written in English. They should be typewritten or computer formatted originals on A4 or letter-size paper with a main (*default*) font type size of at least ten points. Margins and line spacing that allow for reviewer remarks are convenient. Sections must be numbered and must have a title not consisting of formulae only. Long formulae must be displayed and numbered. Special symbols, handwritten or custom-made, must be listed separately.

- *Footnotes*, which should be kept to a minimum and should be brief, must be numbered consecutively; they should be placed in the main text starting at the same page as their footnote mark.
- All *figures* must be numbered; they must be referred to in the text and placed close to the first reference, with the caption at the bottom of the figure.
- *Tables* must be numbered, referred to in the text and placed close to the first reference.
- *References* must be numbered in the order of occurrence in the text. In the text they should be referred to by bracketed numbers. The list of references must be given at the end of the manuscript, i.e. not between the sections and the appendices of the manuscript.

Submission of manuscripts. Manuscripts to be submitted for publication must be previously unpublished, and not under consideration for publication elsewhere. They should be sent, *together with 4 copies*, to the Integration Editorial Office, to the attention of Marion de Vlieger, Delft University of Technology, Department of Electrical Engineering (room 17.26), Mekelweg 4, 2628 CD Delft, The Netherlands. In addition to the original and four copies of the manuscript according to the above preparation section, the following should be provided:

1. A short title of the paper, as specific as possible. Words of which the antonym is highly unlikely in a title (e.g. as "new", "efficient", "correct") should be avoided.
2. An abstract not exceeding 150 words and preferably not containing longer or complicated formulae.
3. A list of keywords.
4. A submission record indicating where the text, either complete or in major part, has been submitted. Also a corresponding author must be indicated, with complete postal address.
5. A passport-size photograph and a short biography of every author.
6. Original drawing of the figures, suitable for photographic reproduction (in large size, on separate sheets, with wide margins).
7. Tables on separate sheets with ample spacing.
8. A separate list of the captions in the *default* font type size.
9. A separate list of references in *default* font type size, preceded by their number between square brackets, and according to the following models:
   - *For a paper in a journal*: J.L. Bentley and D. Wood, An optimal worst case algorithm for reporting intersections of rectangles, IEEE Trans. Comput. C-29 (7) (1980) 53–66.
   - *For a paper in a contributed volume*: H.T. Kung and C.E. Lierson, Algorithms for VLSI processor arrays, in: C.A. Mead and L.A. Conway (Eds.), Introduction to VLSI Systems, Addison-Wesley, Reading, MA, 1980, pp. 271–292.
   - *For an unpublished paper*: H.M. Ahmed, Signal processing algorithm and architectures, Ph.D. Thesis, Dept. of Electrical Engrg. Standford Univ., 1981.

Processing. An *acknowledgement* of the receipt of the manuscript at the Integration Editorial Office will be sent immediately to the address of the corresponding author. If the submission conforms to the above submission rules a *status letter* containing either a decision or a prognosis is sent to the corresponding author within three months after the receipt. If revisions are required a corrected final manuscript according to submission rules given above must be sent to the Integration Editorial Office. Upon acceptance of an article, the author(s) will be asked to transfer *copyright* of the article to the publisher. This transfer will ensure the widest possible dissemination of information. The corresponding author also receives *page proofs*, which should be corrected and returned by airmail to the publishers within three days of receipt. Corrections in the proof stage other than of printer's errors should be avoided: costs arising from such extra corrections will be charged to the author. No page charge is made. 50 offprints of each paper will be provided free of charge. Additional offprints may be ordered at cost.

Submission of electronic text. In order to publish the paper as quickly as possible after acceptance, authors are encouraged to submit the final text also on a 3.5″ or 5.25″ diskette. Both double density (DD) and high density (HD) diskettes are acceptable. The diskette may be formatted with either MS-DOS/PC-DOS or with Macintosh OR. See the Notes for Electronic Text Preparation at the back of this issue for further information. The final manuscript may contain parts (e.g. formulae, complex tables) or last-minute corrections which are not included in the electronic text on the diskette; however, this should be clearly marked in an additional hardcopy of the manuscript. Authors are encouraged to ensure that apart from any such small last-minute corrections, **the disk version and the hardcopy must be identical**. Discrepancies can lead to proofs of the wrong version being made.

Subscription Information
INTEGRATION, the VLSI journal (ISSN 0167-9260). For 1998 volumes 26–27 are scheduled for publication. Subscription prices are available upon request from the publisher. Subscriptions are accepted on a prepaid basis only and are entered on a calendar year basis. Issues are sent by surface mail except to the following countries where air delivery via SAL mail is ensured. Argentina, Australia, Brazil, Canada, China, Hong Kong, India, Israel, Japan, Malaysia, Mexico, New Zealand, Pakistan, Singapore, South Africa, South Korea, Taiwan, Thailand, USA. For all other countries airmail rates are available upon request. Claims for missing issues must be made within six months of our publication (mailing) date.

Orders, claims, and product enquiries: please contact the Customer Support Department at the Regional Sales Office nearest you:

New York: Elsevier Science, PO Box 945, New York, NY 10159-0945, USA; Phone: (+1) (212) 633 3730 [toll free number for North American customers: 1-888-4ES-INFO (437-4636)]; fax: (+1) (212) 633 3680; e-mail: usinfo-f@elsevier.com

Amsterdam: Elsevier Science, PO Box 211, 1000 AE Amsterdam, The Netherlands; phone: (+31) 20 4853757; fax: (+31) 20 4853432; e-mail: nlinfo-f@elsevier.nl

Tokyo: Elsevier Science K.K., 9-15 Higashi-Azabu 1-chome, Minato-ku, Tokyo 106, Japan; phone: (+81) (3) 5561 5033; fax: (+81) (3) 5561 5047; e-mail: info@elsevier.co.jp

Singapore: Elsevier Science, No. 1 Temasek Avenue, #17-01 Millenia Tower, Singapore 039192; phone: (+65) 434 3727; fax: (+65) 337 2230; e-mail: asiainfo@elsevier.com.sg

Rio de Janeiro: Elsevier Science, Rua Sete de Setembro 111/16 Andar, 20050-002 Centro, Rio de Janeiro – RJ, Brazil; phone: (+55) (21) 509 5340; fax: (+55) (21) 507 1191; e-mail: elsevier@campus.com.br [Note (Latin America): for orders, claims and help desk information, please contact the Regional Sales Office in New York as listed above]