Classification Tree Method with Parameter Shielding

Takashi Kitamura[1(✉)], Akihisa Yamada[2], Goro Hatayama[3], Shinya Sakuragi[3], Eun-Hye Choi[1], and Cyrille Artho[4]

[1] National Institute of Advanced Industrial Science and Technology (AIST), Osaka, Japan
{t.kitamura,e.choi}@aist.go.jp
[2] Innsbruck University, Innsbruck, Austria
akihisa.yamada@uibk.ac.at
[3] Omron Social Solutions Co., Ltd., Kyoto, Japan
{goro_hatayama,shinya_sakuragi}@oss.omron.co.jp
[4] KTH Royal Institute of Technology, Stockholm, Sweden
artho@kth.se

Abstract. The Classification Tree Method (CTM) is a structured and diagrammatic modeling technique for combinatorial testing. CTM can express the notion of *"parameter shielding",* the phenomenon that some system parameters become invalidated depending on another system parameter. The current form of CTM, however, is limited in its expressiveness: it can only express parameter shielding that depends on a single parameter. In this paper, we extend CTM with parameter shielding that depends on multiple parameters, proposing CTM_{shield}. We evaluate the proposed extension on several industrial systems. The evaluation finds that parameter shielding often depends on multiple parameters in real systems, and the effectiveness of the extension.

1 Introduction

Testing is an important and often a necessary system development process for assuring system quality in current industrial practice. *Combinatorial testing* is a system testing technique, that effectively tests the interactions of parameters in a system under test. Combinatorial testing derives, typically from specification, a test model, which consists of a list of parameter-values and *constraints* over them. Based on such test models, test suites are designed, that consider various coverage criteria, such as t-way testing [1,2].

Figure 1 shows an example test model, which specifies an IC card system with six parameters, each having two to three values: the *Age* of the card owner, the *Balance* that is already charged in the card, whether *Credit Card (C.C.)* information is available or not, the *Charge Method (C.M.)* and *Charge Amount (C.A.)* the owner specifies to the system, and the *Monthly Total (M.T.)* amount of usage. The model also indicates *constraints* in logic formula, specifying *valid* (and *invalid*) value combinations. The two constraints in the example model specify the following specifications:

S. Tonetta et al. (Eds.): SAFECOMP 2017, LNCS 10488, pp. 230–241, 2017.
DOI: 10.1007/978-3-319-66266-4_15

PARAMETERS	VALUES
Age	child, adult, senior
Balance	>190€, ≤190€
Credit Card (C.C.)	with, without (w/o)
Charge Method (C.M.)	cash, credit card (c.c.)
Charge Amount (C.A.)	10€, 50€
Monthly Total (M.T.)	>390€, ≤390€

CONSTRAINTS
$\neg(C.C. = \text{with} \wedge Age = \text{child})$
$(C.M. = \text{c.c.} \Rightarrow C.C. = \text{with})$

Fig. 1. A test model example for an IC card system; it consists of a parameter-values list (left) and a set of constraints (right).

- "An IC card owned by a child cannot have *Credit Card* information."
- "The *Charge Method* can be by credit card only if a *Credit Card* information is available."

Table 1 shows, as a test suite example, a 2-way test suite of the test model.

Table 1. A 2-way test suite for the test model of Fig. 1, which covers all valid value pairs but avoids invalid ones, e.g. $\langle C.C.{=}with,\ Age{=}child\rangle$, specified by the constraints.

No.	*Age*	*Balance*	*C.C.*	*M.T.*	*C.M.*	*C.A.*
1	child	>190€	w/o	≤390€	cash	10€
2	child	≤190€	w/o	>390€	cash	50€
3	adult	>190€	w/o	≤390€	cash	50€
4	adult	≤190€	with	>390€	c.c	10€
5	senior	>190€	with	≤390€	c.c	50€
6	senior	≤190€	w/o	≤390€	cash	50€
7	senior	>190€	with	>390€	cash	10€

A key challenge in applying combinatorial testing in real-world development is modeling, a. k. a. Input Parameter Modeling [2] or Input Domain Modeling [3]. Models in real-world systems often involve complex constraints on parameter-values. This makes modeling a time-consuming and error-prone task that requires experience and creativity of test experts.

Classification Tree Method (CTM) [4–6] is a structured and diagrammatic approach for the modeling problem. The main characteristic of CTM is that, using a tree-structured modeling language called *Classification Trees (CTs)*, is to be able to describe the notion of "Parameter shielding" concisely, which is a phenomenon that some parameters become invalided (i.e., *shielded*) if some specific values are (or are not) assigned to another parameter.

Suppose, for instance, the following specification SPEC1 is added to the system:

SPEC1: "Charging is allowed only if the *Balance* is below 190€."

Figure 2 shows a CTM model that expresses this specification using a tree structure. The tree structure expresses not only that (1) the relation between parameters and values, but also (2) *compositions* of parameters and (3) parameter shielding. The rounded rectangle node *Charge* combines *Charge Method* and *Charge Amount* of the previous example, and appears under value ≤190€ of *Balance*. This expresses that the two parameters become valid only when the *Balance* is below 190€, and become *invalid* (shielded) otherwise. Table 2 shows a 2-way test suite for the CTM model. Note that some parameters are assigned the *vain* value "—" when they are invalid. Note also that the test suite of Table 1 is not a valid 2-way test suite for the current model anymore, since, e.g., test case No. 1 in Table 1 is not executable under SPEC1.

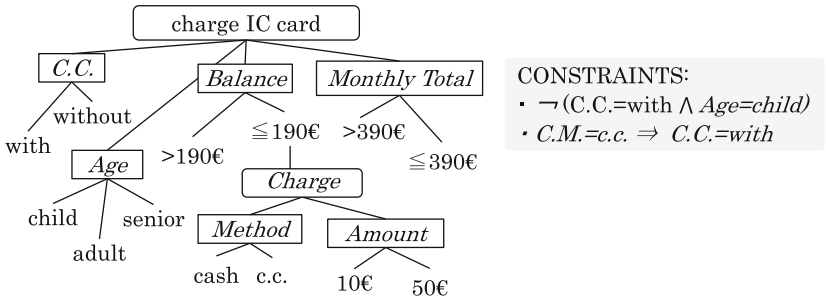


Fig. 2. A CTM test model for the IC card system that expresses SPEC1.

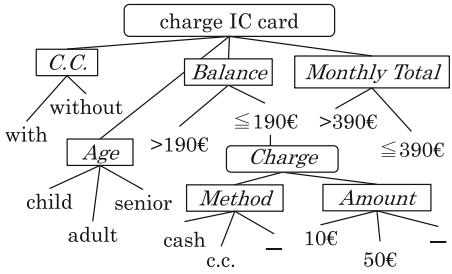
Table 2. A 2-way test suite for the model of Fig. 2 under SPEC1, where parameters *C.M.* and *C.A.* are shielded (as assigned the vain value '—') when *Balance* is >190€.

No.	*Age*	*Balance*	*C.C.*	*M.T.*	*C.M.*	*C.A.*
1	child	≤190€	w/o	≤390€	cash	10€
2	child	≤190€	w/o	>390€	cash	50€
3	child	>190€	w/o	>390€	—	—
4	adult	≤190€	with	>390€	c.c	10€
5	adult	≤190€	w/o	≤390€	cash	50€
6	adult	>190€	with	≤390€	—	—
7	senior	≤190€	with	>390€	cash	10€
8	senior	≤190€	with	≤390€	c.c	50€
9	senior	>190€	w/o	≤390€	—	—

Parameter shielding expressed in a tree structure is a unique and useful feature of CTM; however, its limitation is that it can only describe parameter shielding that depends on a single parameter-value. The reason is obvious: the dependencies of parameter shielding are expressed within the tree structure, and

Table 3. A valid 2-way test suite under SPEC2, where parameters *C.M.* and *C.A.* are shielded when either *Balance* is >190€ or *M.T.* is >390€.

No.	*Age*	*Balance*	*C.C.*	*M.T.*	*C.M.*	*C.A.*
1	child	≤190€	w/o	≤390€	cash	50€
2	child	≤190€	w/o	≤390€	cash	10€
3	child	>190€	w/o	>390€	—	—
4	adult	≤190€	with	≤390€	cash	10€
5	adult	≤190€	with	≤390€	c.c	50€
6	adult	>190€	with	≤390€	—	—
7	senior	≤190€	w/o	>390€	—	—
8	senior	≤190€	w/o	≤390€	c.c	50€
9	senior	≤190€	with	≤390€	cash	10€
10	senior	>190€	with	>390€	—	—



Constraints:
- $\neg$ *(C.C.=with* $\wedge$ *Age=child)*
- *C.M.=c.c.* $\Rightarrow$ *C.C.=with*
- *(M.T.= >390* $\Rightarrow$ *(C.M. =* $-$ $\wedge$ *C.A. =* $-$*))* $\wedge$ *((C.M. =* $-$ $\vee$ *C.A> =*$-$*)* $\Rightarrow$ *M.T.= >390)*

Fig. 3. A CTM test model for the IC card system uner SPEC2, which expresses a test model of the test suite in Table 3.

hence a parameter can only have one parent. In our case studies applying combinatorial testing and CTM to industrial systems, however, we often encountered a demand to express parameter shielding that depends on multiple parameter-values.

For instance, suppose SPEC1 is refined as in the following specification SPEC2:

> SPEC2: "Charging is allowed only if the *Balance* is below 190€ and *Monthly Total* usage is below 390€."

As the node *Charge* should be shielded depending on two (i.e., multiple) parameter-values, this is a typical example of the multi-dependent parameter shielding. Note this time that the test suite of Table 2 is not a valid 2-way test suite anymore, since, e.g., test case No. 2 is not executable under the refined specification SPEC2. A valid 2-way test suite under SPEC2 is as shown in Table 3. Further, it is now difficult to model SPEC2 concisely in CTM. The reason is, as explained, a tree node cannot have multiple parents.

A possible solution that CTM can do is to explicitly handle the vain values and complex constraints involving them as in Fig. 3. However, we assume such manipulations on test models makes themselves too complex and busy, losing conciseness, for engineers to creates and maintain, especially in dealing with industrial-scaled large systems.

In this paper, we propose CTM$_{shield}$, by extending CTM with parameter shielding that can depend on multiple parameter-values, or more generally an arbitrary logic formula. Figure 4 shows the basic idea of the extension, by showing an test model example in CTM$_{shield}$ which expresses SPEC2. Observe that CTM$_{shield}$ is extended with the additional description called *(parameter) shielding conditions*. Observe also that the shielding condition in Fig. 4 specifies SPEC2 directly, using the notation "$P \leftarrow_{shield} V$" to mean parameter-value V shields parameter P. In such a way, we aim to avoid explicitly handling the vain values and complex constraints to express parameter shielding, and thus to retain test models concise and readable. To evaluate the effectiveness of the proposed extension, we conduct experiments via case studies, where we applied combinatorial testing to test industrial systems in the railway domain, using CTM$_{shield}$. As summary, the experimental results showed that parameter shielding was used in 72% of the cases; CTM$_{shield}$ was able to reduce the tree size by 7.13% and the length of constraints by 22.9% on average, compared with CTM of [7,8]. Therefore, CTM$_{shield}$ contributes to saving human effort on modeling.

The paper is organized as follows. Section 2 mentions related studies. Section 3 clarifies the design of CTM$_{shield}$. Section 4 reports our experimental study to evaluate the effectiveness of the proposed extension. Section 5 states our plan for future work.

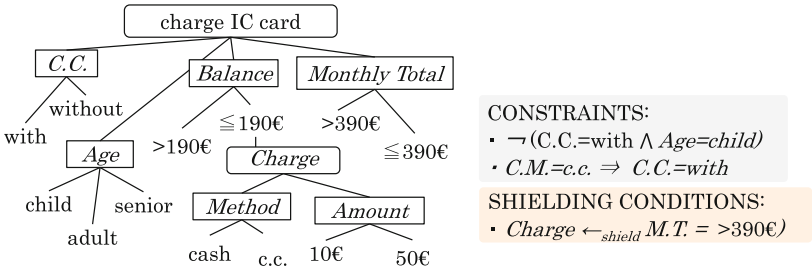


Fig. 4. A CTM$_{shield}$ test model that expresses SPEC2 and hence is a test model of the test suite in Table 3.

2 Related Work

CTM has been recognized as a key technique in the field of combinatorial testing [2], and has been studied on various aspects. For example, Lehmann and Wegener [5] introduced constraints to the original CTM [7] to extend its expressiveness. Prioritizing test cases was studied for effective test design in the setting

of CTM [6]. A test generation algorithm dedicated to CTM was developed in [8]. Also, experimental data in [8] indicate that the structured aspect of CTM reduces the lengths of constraints to be described. For industrial aspects, CTM has been used in industries of safety-critical domains: e. g., it is used in a standard test documentation in automotive industry [9]. Driven by industrial demand, tools to support CTM have been developed by several vendors [10].

The notion of parameter shielding in combinatorial testing has been studied in several different approaches. To our knowledge, the earliest work that is relevant for CTM is by Grochtmann [4]; however, its focus does not seem on parameter shielding but on diagrammatic approaches, as the phenomenon of parameter shielding was not mentioned. Chen et al. [11] first clarified and defined the notion of parameter shielding in the setting of Covering Arrays. which considers only unconstrained and unstructured models. They provided test generation algorithms for this special kind of Covering Arrays. Segall et al. take yet another approach of "common patterns" [12]. They identified several recurring properties in modeling as patterns, which are often hard to capture correctly, and supply solutions for them. The notion of parameter shielding is captured by one of their patterns, called "Conditionally-Excluded-Values" pattern. Zhao et al. developed a test generation tool of combinatorial testing, called CASCADE, which can handle shielding parameters explicitly [13] and its handling mechanism is basically same as the proposed solution in [12].

Our work differs from these works in that our contributions are to propose a modeling language by extending CTM with parameter shielding to advance CTM and evaluate its effectiveness via case studies.

3 Classification Tree Method with Parameter Shielding

This section proposes the modeling language CTM$_{shield}$, which extends CTM with the notion of parameter shielding. To be conscious about the extension, we first define the language for combinatorial testing, next that for CTM, and finally that for CTM$_{shield}$.

The definition of combinatorial testing, whose example is in Fig. 1, is as follows:

Definition 1 (Combinatorial testing). *A combinatorial testing model is a tuple* $m = \langle P, V, \Phi \rangle$*, where* P *is a set of* parameters*,* $V = \{V_p\}_{p \in P}$ *is a family of parameter-values, where* V_p *is the value domain of* p*, and constraints* Φ *are a set of Boolean formula over parameter-values.*

A *test case* is a value assignment to parameters in test model m. Formally, it can be defined as a function $\gamma : P \to V$ such that $\gamma(p) \in V_p$ for every $p \in P$. Note that a test case γ must satisfy all the constraints Φ (noted as $\forall \phi \in \Phi . \gamma \models \phi$ or $\gamma \models \Phi$).

A CTM test model consists of a *Classification Tree (CT)* and constraints. A CT consists of three kinds of nodes: *classifications*, which correspond to parameters in combinatorial testing; *classes*, which correspond to values; and *compositions*, a notion that does not appear in combinatorial testing.

Definition 2 (CTM). *A test model of CTM is a tuple* $m = \langle r, P, V, C, \uparrow, \Phi \rangle \in M$*, where* $\langle P, V, \Phi \rangle$ *forms a test model of combinatorial testing,* C *is a set of compositions,* $r \in C$ *is a root node, and* $\uparrow$ *is a function from* $P \cup C \backslash \{r\}$ *to* $V \cup C$ *that expresses a part of the child-parent relation of the tree structure of CT.*

As some parameters are shielded and assigned the vain value "—", a *test case* of CTM extends that of combinatorial testing as $\gamma : P \to \{—\} \cup V$, while inheriting $\gamma \models \Phi$. A parameter p is shielded, assigned "—", in a test case γ, if its nearest ancestor value is not chosen for the parameter in γ. For example, parameter *C.M.* is shielded in test case No. 3 in Table 2, since its nearest ancestor value ">190€" is not chosen for the parameter *Balance* in the test case.

Now, the definition of CTM$_{shield}$ is given as follows:

Definition 3 (CTM$_{shield}$). *A CTM$_{shield}$ model is a tuple* $m = \langle r, P, V, C, \uparrow, \Phi, \Phi_s \rangle \in M_s$*, where* $\langle r, P, V, C, \uparrow, \Phi \rangle$ *is a CTM model and* Φ_s *is a function from* $P \cup C$ *to Boolean formulas. We denote* $\Phi_s(n)$ *as the* (parameter) shielding condition *of* $n \in P \cup C$*.*

The definition of CTM$_{shield}$ extends that of CTM by the shielding conditions Φ_s. A test case of CTM$_{shield}$ also inherits that of CTM, including the condition of parameter shielding specified by the tree structure. Moreover, in CTM$_{shield}$ a parameter p in a test case is shielded by Φ_s when its shielding condition $\Phi_s(p)$ is satisfied by the test case.

In order to express Φ_s in practice, we take a list of pairs of form $p \leftarrow_{\text{shields}} \phi$ indicating that $\Phi_s(p) = \phi$, and assume $\Phi_s(q) = \textsc{False}$ when q is not specified in the list. Note that a CTM model is a CTM$_{shield}$ model with an empty list of such specifications. Figure 4 is an example of CTM$_{shield}$, and Listing 1.1 shows a formulation of the test model in Fig. 4 according to Definition 3.

Listing 1.1. A formulation of the test model in Fig. 4 according to Definition 3.

```
1  r = {charge IC card (CICC)}
2  C= {Charge}
3  P = {C.C., Age, Balance, Method, Amount, Monthly Total (M.T.) }
4  V_C.C. = {with, without}, V_Amount = {10€, 50€}, V_Age = {child, adult, senior},
5  V_Balance = {>190€, ≤190€}, V_M.T. = {>390€, ≤390€ },
6  V_Method = {cash, c.c.}, V_C.A. = {10€, 50€},
7  ↑ = {(C.C., CICC), (Age, CICC), (Balance, CICC), (Charge, ≤190€), (Method, Charge),
8  (Amount, Charge)}
9  Φ = {¬(C.C. = with ∧ Age = child), C.M. = c.c. ⇒ C.C = with }
10 Φ_s = {(>390€, Charge)}
```

4 Case Studies and Evaluations

This section reports our empirical studies to evaluate the effectiveness of the proposed extension, i.e., CTM$_{shield}$ extending CTM with parameter shielding, through case studies with industrial systems.

We determine the "effectiveness" of CTM$_{shield}$ by the conciseness of test models in CTM$_{shield}$ compared to those in CTM (the one dealt in [5,8]). More specifically, we measure the conciseness of models on their description complexity

Table 4. Summary of experimental results.

	CTM$_{shield}$						CTM					
spec.	P/V	#N	$\lvert\Phi\rvert$	$l(\Phi)$	$\lvert\Phi_s\rvert$	$l(\Phi_s)$	P/V	#N	$\lvert\Phi\rvert$	$l(\Phi)$	$\Delta^N_{\sigma_t}$(%)	$\Delta^\Phi_{\sigma_t}$(%)
e.g.1	$2^5 3^1$	22	3^2	6	3^1	3	$2^3 3^2 4^1$	24	$3^2 11^1$	17	8.3	47.1
A-1	$2^{14} 6^1$	58	3^3	9	4^2	8	$2^{10} 3^4 6^1$	62	$3^3 13^2$	35	6.3	51.4
A-2	$2^{15} 3^3 4^1 5^1$	76	3^2	6	4^1	4	$2^{14} 3^4 4^1 5^1$	77	$3^3 4^1$	13	1.3	23.1
A-3	$2^{23} 3^2$	93	3^8	24	$3^1 4^1$	7	$2^{18} 3^7$	98	$3^8 4^1 15^1$	43	5.1	27.9
A-4	$2^{19} 3^1 4^2$	103	3^4	12	4^3	12	$2^{15} 3^5 4^2$	107	$3^4 4^2 13^1$	33	3.7	27.3
A-6	2^{18}	71	3^8	24	4^6	24	$2^{10} 3^8$	79	$3^8 4^3 13^1 15^1$	64	10.0	25.0
A-5	$2^{19} 3^4 5^2 6^1 10^1$	112	3^{17}	51	4^4	16	$2^{16} 3^6 4^1 5^2 6^1 10^1$	116	$3^{17} 4^2 13^2$	85	3.4	21.2
A-8	$2^6 3^1$	27	0	0	4^2	8	$2^4 3^3$	29	4^2	8	6.7	0
A-7	$2^{17} 3^3 9^1 13^1 15^1 33^1$	159	0	0	4^7	28	$2^{12} 3^8 9^1 14^1 16^1 33^1$	169	4^7	28	4.1	0
A-9	$2^7 5^1$	36	3^4	12	3^1	3	$2^6 3^1 5^1$	37	3^5	15	2.6	0
A-10	$2^9 3^1$	39	3^1	3	5^1	5	$2^8 3^2$	40	$3^1 5^1$	8	2.4	0
A-11	$2^{10} 3$	46	3^1	3	4^1	4	$2^9 3^3$	47	$3^1 4^1$	7	2.1	0
A-12	$2^{37} 7^1$	139	3^4	12	4^3	12	$2^{34} 3^3 7^1$	142	$3^4 4^3$	24	2.1	0
A-13	$2^9 3^1$	31	0	0	0	0	$2^9 3^1$	31	0	0	0	0
A-14	$2^{16} 3^3$	67	0	0	0	0	$2^{16} 3^3$	67	0	0	0	0
A-15	$2^{10} 3^1$	42	0	0	0	0	$2^{10} 3^1$	42	0	0	0	0
A-16	$2^7 3^1$	25	0	0	0	0	$2^7 3^1$	25	0	0	0	0
A-17	$2^3 8^1$	23	0	0	0	0	$2^3 8^1$	23	0	0	0	0
A-18	2^3	17	0	0	0	0	2^3	17	0	0	0	0
A-19	2^{15}	58	0	0	0	0	2^{15}	58	0	0	0	0
										avg. of system A	4.2	14.7
B-1	$1^2 2^{17} 4^2$	65	$1^2 2^8 3^9 4^2$	0	4^5	20	$1^2 2^8 3^9 4^2$	87	$4^5 25^1 27^1$	72	10.2	72.2
B-2	$1^3 2^{43} 4^6$	165	$1^3 2^{16} 3^{27} 4^6$	0	$4^{17} 7^1$	75	$1^3 2^{16} 3^{27} 4^6$	220	$4^{15} 25^2 69^1$	179	12.2	58.1
B-3	$2^5 3^7$	43	$2^2 3^5 4^5$	18	4^4	16	$2^2 3^5 4^5$	58	$3^6 4^2 15^1 17^1$	58	13.6	41.4
B-4	$2^6 3^8$	50	$2^3 3^5 4^6$	27	4^5	20	$2^3 3^5 4^6$	63	$3^9 4^3 15^1 17^1$	71	14.1	33.8
B-5	$2^6 3^7 4^1$	51	$2^3 3^5 4^5 5^1$	33	4^5	20	$2^3 3^5 4^5 5^1$	64	$3^{11} 4^3 15^1 17^1$	77	13.8	31.2
B-6	2^{11}	33	$2^{10} 3^1$	0	4^1	4	$2^{10} 3^1$	49	4^1	4	2.0	0
										avg. of system B	11.0	39.4
										total avg.	7.13	22.9

in terms of the number of parameter-values and the length of constraints needed to describe the models. Our evaluation poses the following three research questions:

RQ1: How often is the parameter shielding condition used? What is its usage rate?
RQ2: How many tree nodes can CTM$_{shield}$ reduce, compared with CTM?
RQ3: How much length of constraints can CTM$_{shield}$ reduce, compared with CTM?

4.1 Setting

In the case studies, we applied combinatorial testing to functional testing of 25 system-level functions of the following two distinct industrial systems in railway domain, and in doing so we used CTM$_{shield}$ for modeling the functions: 19 functions from a ticket gate system (system A) and 7 functions from a payment system (system B).

For comparison between CTM$_{shield}$ and CTM, we also prepared test models in CTM. We prepare a program to translate test models in CTM$_{shield}$ to equivalent ones in CTM; i.e., it handles complex manipulation of constraints and dummy nodes. For example, it inputs the CTM$_{shield}$ test model in Fig. 4, and outputs the CTM test model in Fig. 3.

Next, we measure the following metrics of the test models in CTM$_{shield}$ and CTM:

1. P/V: The size of parameter-values; this is expressed as $g_1^{k_1} g_2^{k_2} ... g_n^{k_n}$, which means that for each i there are k_i parameters that have g_i values, following [14,15].
2. $\#N$: The number of nodes; this is the summation of the numbers of compositions, parameters, and values.
3. $l(\phi)$: The length of constraints ϕ, which is defined in a similar way as [16] as follows: $l(a) = 1$ for all atoms a, $l(\neg P) = 1 + l(P)$, and $l(P * Q) = 1 + l(P) + l(Q)$ where $*$ is a binary operator of $\wedge, \vee, \Rightarrow$, and $\Leftrightarrow$. E.g., $l(\underline{\neg_1 P_2 = v_{1_2}} \underline{\vee_3 P_3 = v_{1_4}}) = 4$.

Since CTM$_{shield}$ has an additional description component of parameter shielding conditions, we also measure the following two metrics for test models of CTM$_{shield}$:

4. $l(\Phi_s)$: The length of parameter shielding conditions Φ_s. As mentioned in Sect. 3 and exemplified in Fig. 4, a shielding condition is expressed using "$\leftarrow_{shield}$" in practice; e. g., "$P_1 \leftarrow_{shield} \neg P_2 = v_1 \vee P_3 = v_1$" to mean $\Phi_s(P_1) = \neg P_2 = v_1 \vee P_3 = v_2$", where P_1, P_2, P_3 expresses parameters and v_1 and v_2 values. Thus, we define the length of a shielding condition for parameter n by $l(\Phi_s(n)) + 2$, regarding $\leftarrow_{shield}$ as a binary operator. For example, $l(\Phi_s(P_1)) = l(\underline{\neg_1 P_2 = v_{1_2}} \underline{\vee_3 P_3 = v_{1_4}}) + 2 = 6$
5. $|\Phi_s|$: The size of Φ_s in the form $1^{k_1} 2^{k_2} ... n^{k_i}$, which this time means there are k_i conditions whose length is n for each $n \in Nat$ while n^{k_i} are omitted if $k_i = 0$.

In order to quantitatively answer the research questions, from data about the test model of CTM (m_c) and that of CTM (m_s) for each function, we retrieve the following:

– the reduction rate of the number of nodes $\Delta^N_{\%}$:

$$\Delta^N_{\%} =: \frac{\#N^{m_c} - \#N^{m_s}}{\#N^{m_c}} \tag{1}$$

– the reduction rate of constraint length $\Delta^{\Phi}_{\%}$:

$$\Delta^{\Phi}_{\%} =: \frac{l(\Phi^{m_c}) - (l(\Phi^{m_s}) + l(\Phi_s^{m_s}))}{l(\Phi^{m_c})} \tag{2}$$

where $\#N^m$, $l(\Phi^m)$, and $l(\Phi_s^m)$ respectively mean the number of nodes, length of constraints, length of shielding conditions in test model m.

Note that $\Delta_{\%}^{\Phi}$ considers not only the length of constraints, but also the length of shielding conditions for test model in CTM$_{shield}$ m_s. This is for a fair comparison. We expect (and will see) CTM$_{shield}$ can in fact reduce the length of the constraints. However, this is achieved at the cost of describing the shielding conditions. To avoid such an unfair comparison, we designed in $\Delta_{\%}^{\Phi}$ to consider not only constraints length but also the length of shielding conditions in test models of CTM$_{shield}$.

4.2 Results and Observations

Table 4 summarizes the experimental results. The first column shows retrieved data from the test model example in CTM$_{shield}$ in Fig. 4 and an equivalent test model in CTM in Fig. 3, whose main points are read as follows:

1. The CTM$_{shield}$ test model in Fig. 4 has five parameters for two values and one parameter for three values, hence its P/V is expressed as $2^5 3^1$. On the other hand, the CTM test model in Fig. 3 has three parameters for two values, two for three values, and one for four values, hence its P/V is expressed as $2^3 3^2 4^1$.
2. The number of nodes in the CT in CTM is 24, while that in CTM$_{shield}$ is 22. Thus, the number of nodes is reduced by 2 ($= 24 - 22$), and the node reduction rate ($\Delta_{\%}^{N}$) is 8.3%($= \frac{2}{24}$).
3. The constraint length of the CTM test model is 11, while that of CTM$_{shield}$ is 6. We also take the description cost of shielding conditions for CTM$_{shield}$ into account, as the length of shielding conditions which is 3. Thus, according to the definition, the reduction rate of constraint length ($\Delta_{\%}^{\Phi}$) is 47.1%($= \frac{(17-(3+6))}{17} = \frac{8}{17}$).

From the summary of the experimental results shown in Table 4, we answer the research questions as follows:

- **Answer for RQ1:** The shielding conditions were not necessarily used for all the cases; instead, they are used in 12 out of 19 cases (63.1%) for system A and in all the six cases (100%) for system B; hence 72% ($= 18/25$) in total of systems A and B.
- **Answer for RQ2:** For the cases where parameter shielding are used, the reduction rate of the number of tree nodes by CTM$_{shield}$ ($\Delta_{\%}^{N}$) is on average 4.2% for system A and 11.0% for system B; 7.13% on the total average.
- **Answer for RQ3:** For the cases where parameter shielding are used, CTM$_{shield}$ reduces the constraint length, compared with CTM, (i.e., $\Delta_{\%}^{\Phi}$) on average by 14.7% for system A, by 39.4% for system B; by 22.9% on the total average.

Note that all the test models for the functions in both systems are expressed as trees, from which we can consider structured and diagrammatic modeling approach of CTM is useful and effective in practice. Also, CTM$_{shield}$ shows a

higher effectiveness in system B than system A, from which we may consider that the effectiveness of using CTM_{shield} differs between systems. As shorter and simpler constraints reduce the human effort on modeling, we consider CTM_{shield} to be effective in real-world settings.

5 Conclusion and Future Work

This work tackled a modeling problem in combinatorial testing, which is a main concern for its use in real developments. We extend CTM, which have been studied and used as a practical modeling language in combinatorial testing, with parameter shielding, and proposed CTM_{shield}. Our experiments via case studies confirmed its effectiveness.

We plan to conduct more empirical studies to evaluate the effectiveness of CTM_{shield} when used to model industrial systems. We leave to future work a theoretical analysis of the proposed extension, such as consistency arguments with different formalisms for parameter shielding [11], theoretical analysis of effectiveness of the extension such as the maximum reduction of the number of nodes, the size of constraints per shielding condition, etc. We also plan to extend our combinatorial testing tool CALOT [8,17,18] with this feature of parameter shielding.

Acknowledgement. This work is partly supported by JST A-STEP grant AS2524001H.

References

1. Kuhn, D.R., Wallace, D.R., Gallo, A.M.: Software fault interactions and implications for software testing. IEEE Trans. Softw. Eng. **30**(6), 418–421 (2004)
2. Kuhn, R.D., Kacker, R.N., Lei, Y.: Introduction to Combinatorial Testing. CRC Press, Boca Raton (2013)
3. Ammann, P., Offutt, J.: Introduction to Software Testing. Cambridge University Press, Cambridge (2008)
4. Grochtmann, M.: Test case design using classification trees. In: Proceedings of STAR 1994 (1994)
5. Lehmann, E., Wegener, J.: Test case design by means of the CTE XL. In: Proceedings of EuroSTAR (2000)
6. Kruse, P.M., Luniak, M.: Automated test case generation using classification trees. Softw. Qual. Prof. **13**(1), 4–12 (2010)
7. Grochtmann, M., Wegener, J.: Test case design using classification trees and the classification-tree editor CTE. In: Proceedings of the of QW 1995 (1995)
8. Kitamura, T., Yamada, A., Hatayama, G., Artho, C., Choi, E.H., Do, N.T.B., Oiwa, Y., Sakuragi, S.: Combinatorial testing for tree-structured test models. In: Proceedings of QRS 2015, pp. 141–150. IEEE CPS (2015)
9. MODISTARC: OSEK/VDX operating system test plan version 2.0 (1999). http://www.osek-vdx.org/. Accessed 07 Mar 2016
10. Berner & Mattner: TESTONA ver.5.1.2 http://www.testona.net/en

11. Chen, B., Yan, J., Zhang, J.: Combinatorial testing with shielding parameters. In: Proceedings of APSEC 2010, pp. 280–289. IEEE CPS (2010)
12. Segall, I., Tzoref-Brill, R., Zlotnick, A.: Common patterns in combinatorial models. In: 2012 Proceedings of ICST 2012, Montreal, pp. 624–629. IEEE CPS (2012)
13. Zhao, Y., Zhang, Z., Yan, J., Zhang, J.: CASCADE: A test generation tool for combinatorial testing. In: Proceedings of ICSTW 2013, pp. 267–270. IEEE CPS (2013)
14. Cohen, D.M., Dalal, S.R., Fredman, M.L., Patton, G.C.: The AETG system: An approach to testing based on combinatiorial design. IEEE Trans. Softw. Eng. **23**(7), 437–444 (1997)
15. Garvin, B.J., Cohen, M.B., Dwyer, M.B.: An improved meta-heuristic search for constrained interaction testing. In: Proceedings of SSBSE 2009, pp. 13–22 (2009)
16. Büning, H.K., Lettmann, T.: Propositional Logic Deduction and Algorithms. Cambridge University Press, Cambridge (1999)
17. Yamada, A., Kitamura, T., Artho, C., Choi, E.H., Oiwa, Y., Biere, A.: Optimization of combinatorial testing by incremental SAT solving. In: Proceedings of ICST 2015, pp. 1–10. IEEE CPS (2015)
18. Yamada, A., Biere, A., Artho, C., Kitamura, T., Choi, E.: Greedy combinatorial test case generation using unsatisfiable cores. In: Proceedings of ASE 2016, pp. 614–624. IEEE CPS (2016)