

# 소프트웨어 컴포넌트 기반 로봇 시스템을 위한 입출력 연관관계 기반 적응형 조합 테스트 기법

## Input/Output Relationship Based Adaptive Combinatorial Testing for a Software Component-based Robot System

강정석, 박홍성\*

(Jeong Seok Kang<sup>1</sup> and Hong Seong Park<sup>1,\*</sup>)

<sup>1</sup>The Department of Electronic and Communication Engineering, Kangwon National University

**Abstract:** In the testing of a software component-based robot system, generating test cases for the system is a time-consuming and difficult task that requires the combining of test data. This paper proposes an adaptive combinatorial testing method which is based on the input/output relationship among components and which automatically generates the test cases for the system. The proposed algorithm first generates an input/output relationship graph in order to analyze the input/output relationship of the system. It then generates the reduced set of test cases according to the analyzed type of input/output relationship. To validate the proposed algorithm some comparisons are given in terms of the time complexity and the number of test cases.

**Keywords:** software component based robot system, component based development, combinatorial testing, input/output relationship, test case

### I. 서론

로봇 기술이 다양해지고 시스템의 규모와 복잡도가 증가함에 따라 이를 해결하기 위한 방법으로 컴포넌트 기반 개발(component-based development)이 대두되고 있다[1-6]. 컴포넌트 기반 개발은 기존 개발 방식과는 달리 특정 프레임워크 상에서 실행되는 부품화된 컴포넌트를 바탕으로 이를 조합하여 더 큰 컴포넌트를 만들거나 애플리케이션을 개발하는 새로운 기법이다.

컴포넌트 기반 개발 방법론을 통해 개발된 소프트웨어 컴포넌트 기반 로봇 시스템(SCRS: Software Component-based Robot System)은 재사용 가능한 여러 컴포넌트들로 구성되며, 각 컴포넌트들은 입출력 인터페이스를 통해서만 연결되고 일부 입출력은 시스템의 외부로 노출된다. 그리고 입출력 연관관계 관점에서 SCRS는 각 컴포넌트의 내부 입출력 연관관계(Intra I/O relationship)와 컴포넌트들간 입출력 연관관계(Inter I/O relationship)를 가지며, 컴포넌트의 내부 입출력 연관관계는 각 컴포넌트 개발자에 의해 정의되고 컴포넌트들간 입출력 연관관계는 각 컴포넌트들을 조합하여 시스템을 개발하는 개발자에 의해 정의된다.

소프트웨어 품질관리에 있어 가장 널리 이용되고 있는 기법은 소프트웨어 테스트이며, 소프트웨어 개발에 소요되

는 총 비용의 50% 이상과 총 시간의 50% 정도가 개발된 소프트웨어의 테스트 작업에 할당되고 있고, SCRS의 정확한 동작을 검증하기 위한 테스트 케이스를 수동으로 생성하는 것은 많은 시간과 비용이 소비된다[7-10]. 각 입력의 테스트 데이터들을 서로 조합하여 테스트 케이스를 생성할 때 모든 조합의 수를 테스트 하는 것(exhaustive testing)은 테스트 케이스가 폭발적(explosion)으로 늘어나 테스트 시 많은 시간과 비용이 소비된다. 조합 테스트 기법(Combinatorial testing) [11-14]은 각 입력들간 교호작용(interaction)으로 인해 발생하는 결함(fault)을 찾을 수 있는 테스트 기법으로 불필요한 테스트 데이터 조합을 제거함으로써 테스트 케이스의 수를 줄인다. 하지만 기존의 조합 테스트 기법들은 테스트 대상 시스템의 입력들 간의 교호작용을 2-way 혹은 3-way 방식 등으로 미리 가정하여, 축소된 테스트 케이스 집합을 생성한다. 실제 SCRS 입력들 간의 교호작용은 가정과 다를 수 있기 때문에 불필요한 테스트 케이스가 존재하거나 결함을 발견할 수 있는 테스트 케이스가 존재하지 않을 수도 있다. 이러한 문제를 해결하고자 기존의 입출력 연관관계 기반 조합 테스트 기법들[15-17]은 실제 입출력 연관관계를 고려하여 각 테스트 데이터들을 조합하는 방법을 제안하였다. 기존의 기법들[15-17]은 실제 입력들 간의 연관관계를 사전에 가정하고 있으며, 연관관계에 따라 불필요한 테스트 케이스를 생성하거나 시간 복잡도가 복잡하다는 단점을 가지고 있다. 그러므로 SCRS의 실제 입출력 연관관계를 고려하여 각 입력 파라미터들의 테스트 데이터를 조합함으로써 시간 복잡도를 간소화 하고 불필요한 테스트 케이스를 줄이기 위한 방법이 필요하다.

본 논문에서는 소프트웨어 컴포넌트 기반 시스템을 위한 입출력 연관관계 기반 적응형 조합 테스트 기법을 제안한

\* Corresponding Author

Manuscript received January 12, 2015 / revised February 13, 2015 / accepted April 16, 2015

강정석, 박홍성: 강원대학교 전자통신공학과

(jasonkangok@gmail.com/hspark@kangwon.ac.kr)

※ 본 논문은 2014년도 강원대학교 학술연구조성비 (과제번호-120141520)와 산업통상자원부의 산업융합원천기술개발사업비로 연구하였음.

다. 제안하는 기법은 테스트 관점에서 SCRS의 입출력 연관관계 구조를 정의하고, SCRS의 각 컴포넌트의 내부 입출력 연관관계와 컴포넌트들 간 입출력 연관관계를 기반으로 시스템의 입출력 연관관계 집합을 생성하는 방법을 제안한다. 그리고 SCRS의 입출력 연관관계 구조에 따라 세 가지의 조합 기법들을 통해 적응적으로 테스트 데이터를 조합하여 테스트 케이스를 생성한다. 본 논문은 시스템을 구성하는 단위 컴포넌트들간의 입출력 연관관계는 사전에 주어지고, 테스트 대상 시스템의 동작은 결정론적(deterministic)이라고 가정한다.

본 논문은 다음과 같이 구성된다. 제 II 장에서는 테스트 관점에서 소프트웨어 컴포넌트 기반 시스템을 정의하고 세 가지 타입의 입출력 연관관계 구조를 설명한다. 제 III 장에서는 제안하는 입출력 연관관계 기반 적응형 조합 테스트 기법을 소개하고, 제 IV 장에서는 제안하는 기법의 효율성을 검증하기 위한 몇 가지 실험을 설명한다. 마지막으로 제 V 장에서는 논문을 요약하고 결론을 맺는다.

**II. SCRS용 테스트 모델 및 입출력 연관관계 구조**

소프트웨어 컴포넌트 기반 로봇 시스템(SCRS)은 그림 1의 예제에서 보는 바와 같이 여러 단위 컴포넌트들로 구성되며, 각 단위 컴포넌트들은 입출력을 통해 서로 연결되어 동작하거나 독립적으로 동작한다[18].

CC는 SCRS를 구성하는 모든 단위 컴포넌트들의 집합을 의미하며,  $C_i$ 는 시스템을 구성하는  $i$  번째 단위 컴포넌트를 의미한다.  $C_{in}^i$ 과  $C_{out}^i$ 는 각각  $C_i$  컴포넌트의 모든 입력 파라미터 집합과 모든 출력 파라미터 집합을 의미하며,  $x_{i,j}$ 와  $y_{i,k}$ 는 각각  $C_{in}^i$ 과  $C_{out}^i$ 에 속한 입출력 파라미터로서  $C_i$  컴포넌트의  $j$  번째 입력 파라미터와  $k$  번째 출력 파라미터를 말한다. 그리고  $EXT_{in}(C_i)$ 와  $EXT_{out}(C_i)$ 는 각각  $C_i$  컴포넌트의 입출력 파라미터들 중에서 SCBS 외부로 노출되는 외부 입출력 파라미터들의 집합을 의미하며,  $CC_{in}$ 과  $CC_{out}$ 은 각각 SCBS 외부로 노출되는 모든 입출력 파라미터들의 집합으로서 모든 컴포넌트들의  $EXT_{in}(C_i)$ 와  $EXT_{out}(C_i)$ 의 합집합이 된다. 시스템의 입출력 연관관계 관점에서  $RP_{intra}$ 는 각 단위 컴포넌트들의 내부 입출력 연관관계 집합인  $C_{rp}^i$ 의 합집합,  $RP_{inter}$ 는 시스템을 구성하는 각 단위 컴포넌트들간의 외부 입출력 연관관계 집합을 의미하며,  $RP$ 는  $RP_{intra}$ 와  $RP_{inter}$ 의 합집합을 의미한다.

앞선 정의들에 의해 소프트웨어 컴포넌트 기반 시스템 SCRS는 다음과 같이 정의 될 수 있다.

$$SCRS = (CC, RP) \tag{1}$$

여기서,  $CC = \bigcup_{i=1}^k C_i$ ,  $RP = RP_{intra} \cup RP_{inter}$ 이며,  $C_{in}^i$ ,  $C_{out}^i$ ,  $C_{rp}^i$ ,  $CC_{in}$ ,  $CC_{out}$ ,  $RP_{intra}$  그리고  $RP_{inter}$ 는 다음과 같다.  $C_{in}^i = \{x_{i,1}, x_{i,2}, \dots, x_{i,m}\}$ ,  $C_{out}^i = \{y_{i,1}, y_{i,2}, \dots, y_{i,m}\}$ ,  $C_{rp}^i = C_{in}^i \times C_{out}^i$ ,  $CC_{in} = \bigcup_{C \in CC} EXT_{in}(C_i)$ ,  $CC_{out} = \bigcup_{C \in CC} EXT_{out}(C_i)$ ,  $RP_{intra} = \bigcup_{i=1}^k C_{rp}^i$ ,  $RP_{inter} = \bigcup_{i,j} (C_{out}^i \times C_{in}^j)$ .

SCRS를 테스트하기 위해서는  $CC_{in}$ 에 속한 각 입력 파라

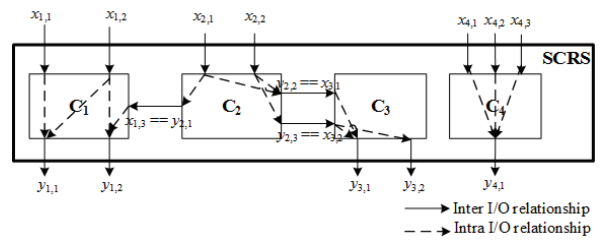


그림 1. SCRS의 컴포넌트 구성 예.

Fig. 1. The example of architecture of SCRS.

미터에 대한 테스트 데이터가 필요하다.  $D(x_i)$ 는 입력 파라미터  $x_i$ 에 대한 테스트 데이터 집합을 의미하며  $D(x_i) = \{d_1, d_2, \dots, d_k\}$ 으로 정의된다. 시스템의 전체 테스트 케이스 집합  $T_m$ 은 다음과 같이  $CC_{in}$ 에 속한 각 입력 파라미터  $x_i$ 의 테스트 데이터 집합( $D(x_i)$ )의 모든 조합으로 정의되고,  $p$ -tuples( $d_1, d_2, \dots, d_p$ )로 구성된다. 여기서  $p$ 는  $CC_{in}$ 의 크기 ( $|CC_{in}|$ )이다. 그리고  $T_m$ 의 크기( $|T_m|$ )는 각 입력 파라미터들의 테스트 데이터 집합 크기( $|D(x_i)|$ )의 곱으로 계산된다.

$$T_m = D(x_1) \times D(x_2) \times \dots \times D(x_p), \tag{2}$$

$$|T_m| = |D(x_1)| * |D(x_2)| * \dots * |D(x_p)|. \tag{3}$$

SCRS를 테스트 시 각 입력 파라미터의 모든 테스트 데이터에 대한 조합으로 생성한 테스트 케이스 집합( $T_m$ )의 크기가 클수록 테스트 하는데 드는 시간 및 비용이 증가하게 된다. 본 논문에서는  $T_m$ 을 커버(cover)할 수 있는 축소된 테스트 케이스 집합  $T_m^{red}$ 을 생성한다. 여기서 "커버(cover)"의 의미는 축소된 테스트 케이스 집합만으로 전체 테스트 케이스 집합을 표현할 수 있다는 것이다.

시스템의 외부 입출력 관점에서 살펴보면, 시스템의 외부 입력 파라미터 집합은  $EXT_{in}(C_i) \subset CC_{in}$  관계를 가지므로  $EXT_{in}(C_i)$ 에 속한 각 입력 파라미터  $x_j$ 의 테스트 데이터 집합( $D(x_j)$ )은 시스템의 외부 입력 파라미터에 대한 테스트 데이터 집합이 될 수 있다. 각 테스트 데이터 집합을 포함하는 외부 입력 파라미터 집합  $F = \{f_1, f_2, \dots, f_n\}$ 로 정의되며,  $n = |CC_{in}|$ 이 된다. 그리고 집합  $F$ 에 속한 각 외부 입력 파라미터  $f_i = (x_i, D(x_i))$ 로 구성되며, 이때  $x_i \in CC_{in}$ 이다.  $D(f_i)$ 는 외부 입력 파라미터  $f_i$ 에 속한  $D(x_i)$ 를 의미한다.  $X(y_i)$ 는 외부 출력 파라미터  $y_i$ 에 영향을 주는 외부 입력 파라미터들의 집합을 의미하고, 각 외부 입력 파라미터들간의 연관관계 집합  $R = \{r_1, r_2, \dots, r_m\}$ 로 정의되며, 집합  $R$ 에 속한 각 연관관계  $r_i = \{f_1, f_2, \dots, f_k\}$ 는 외부 입력 파라미터들로 구성된다.

각 외부 입력 파라미터들간의 연관관계( $R$ )은 그림 2에서 보는 바와 같이 크게 세 가지 타입, 모든 연관관계(그림 2(a)), 독립적 연관관계(그림 2(b)), 교집합 연관관계(그림 2(c))로 정의된다.

모든 연관관계는 그림 2(a)에서 보는 바와 같이 특정 외부 출력에 대해 모든 외부 입력이 영향을 주는 경우이다. 이때 각 외부 출력에 영향을 주는 외부 입력들의 집합을 구하면  $X(y_1) = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$ ,  $X(y_2) = \{x_2, x_3, x_4\}$ ,  $X(y_3) = \{x_4, x_5, x_7\}$ 이 되며,  $X(y_2)$ 와  $X(y_3)$ 는  $X(y_1)$ 의 부분 집합이므로 삭제되고, 연관관계 집합  $R = \{r_1\}$ ,  $r_1 = \{x_1, x_2, x_3, x_4,$

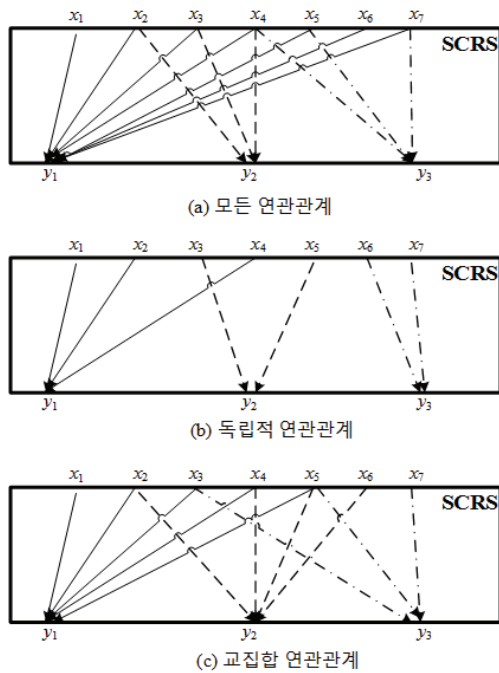


그림 2. SCRS의 입출력 연관관계 구조.  
Fig. 2. The types of input/output relationships of SCRS.

$x_5, x_6, x_7$ 이 된다. 즉, 모든 연관관계 구조에서 집합  $R$ 의 크기는 1이 된다.

독립된 연관관계는 그림 2(b)에서 보는 바와 같이 각 외부 출력에 서로 다른 외부 입력이 영향을 주는 경우이다. 이때 각 외부 출력에 영향을 주는 외부 입력들의 집합을 구하면  $X(y_1)=\{x_1, x_2, x_3\}$ ,  $X(y_2)=\{x_4, x_5\}$ ,  $X(y_3)=\{x_6, x_7\}$ 이 되며, 연관관계 집합  $R=\{r_1, r_2, r_3\}$ ,  $r_1=\{x_1, x_2, x_3\}$ ,  $r_2=\{x_4, x_5\}$ ,  $r_3=\{x_6, x_7\}$ 이 되며 각 연관관계는 서로 독립적이다.

교집합 연관관계는 그림 2(c)에서 보는 바와 같이 외부 입력이 몇몇 출력에 동시에 영향을 주는 경우이다. 이때 각 외부 출력에 영향을 주는 외부 입력들의 집합을 구하면  $X(y_1)=\{x_1, x_2, x_3, x_4, x_5\}$ ,  $X(y_2)=\{x_2, x_4, x_5, x_6\}$ ,  $X(y_3)=\{x_3, x_5, x_7\}$ 이 되며, 연관관계 집합  $R=\{r_1, r_2, r_3\}$ ,  $r_1=\{x_1, x_2, x_3, x_4, x_5\}$ ,  $r_2=\{x_2, x_4, x_5, x_6\}$ ,  $r_3=\{x_3, x_5, x_7\}$ 이 되며 각 연관관계는 서로 교집합이 존재한다.

III. 입출력 연관관계 기반 적응형 조합 테스트 기법

제안하는 입출력 연관관계 기반 적응형 조합 테스트 기법은 그림 3에서 보는 바와 같이 입력으로 SCRS의 입출력 연관관계 정보  $RP$ , SCRS의 외부 입력 집합  $CC_{in}$ , SCRS의 외부 출력 집합  $CC_{out}$ 을 가지며, 출력으로는 SCRS를 테스트하기 위한 축소된 테스트 케이스 집합  $T_{in}^{red}$ 을 생성한다.

먼저 SCRS에 대한 입출력 연관관계 그래프(IORG: Input/Output Relationship Graph)를 생성하고, 명세 기반 테스트 기법들을 이용하여 SCRS의 각 외부 파라미터마다 테스트 데이터를 생성하여 테스트 데이터를 포함하는 외부 입력들의 집합  $F$ 를 생성하고, IORG를 이용하여 입력들 간의 연관관계 집합  $R$ 을 생성한다. 그런 다음 연관관계 집합  $R$ 의 크기가 1이라면  $F$ 에 속한 모든 외부 입력들을 모두 조

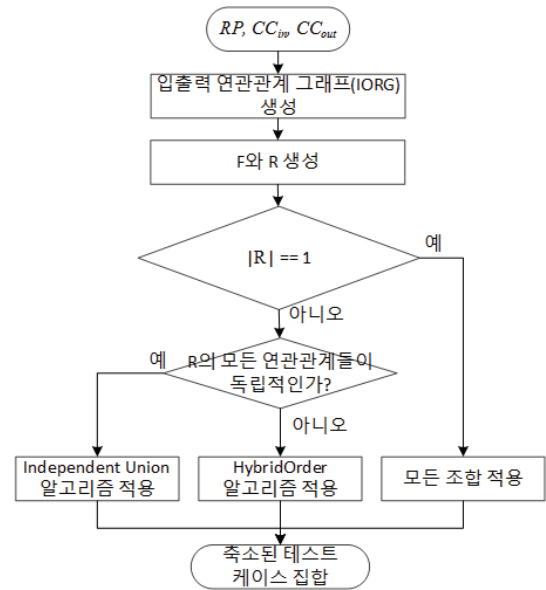


그림 3. 입출력 연관관계 기반 적응형 조합 테스트 기법.  
Fig. 3. The overview of input/output relationship based adaptive combinatorial testing.

합하여 테스트 케이스를 생성한다. 그렇지 않고 집합  $R$ 에 속한 연관관계가 서로 독립적이라면 본 논문에서 제안하는 Independent Union 기법을 통해 테스트 데이터를 조합하고, 집합  $R$ 에 속한 연관관계들이 서로 교집합에 존재한다면 본 논문에서 제안하는 HybridOrder 기법을 통해 테스트 데이터를 조합하여 테스트 케이스를 생성한다.

1. 입출력 연관관계 집합( $R$ ) 생성

입출력 연관관계 그래프(IORG)는 SCRS의 모든 입출력 연관관계를 표현하는 데이터 구조로써 다음과 같이  $IORG=(V,E)$ 로 구성된다. 여기서  $V$ 는 그래프상의 정점으로써 모든 단위 컴포넌트들의 입출력 파라미터이며,  $E$ 는 그래프상의 정점들 간의 에지(edge)로써 각 입출력 파라미터들의 연관관계를 나타낸다. 입출력 연관관계 그래프는 [18]에서 제안한 기법을 이용하여 생성한다. 그림 4는 그림 1의 예제 SCRS를 입출력 연관관계 그래프 생성 기법을 통해 생성한 IORG이다. 그림 4에서 보는바와 같이 SCRS의 IORG는 10개의 입력 노드와 8개의 출력노드, 그리고 각 노드들 간의 에지들로 구성된다.

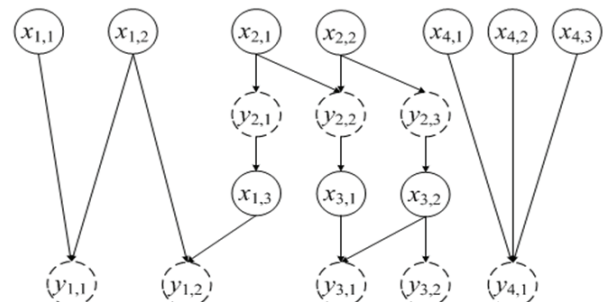


그림 4. 예제 SCRS의 입출력 연관관계 그래프.  
Fig. 4. The input/output relationship graph of the example SCRS.

IORG가 생성된 후 SCRS의 각 외부 입력들의 테스트 데이터를 명세 기반 테스트 기법들[19,20]을 이용하여 생성한 후 집합  $F$ 를 생성한다. 그리고 각 외부 출력 파라미터에 영향을 주는 부분 집합을 포함하지 않는 외부 입력 파라미터들의 집합을 구하여 집합  $R$ 을 생성한다.

## 2. 적응적 테스트 케이스 생성

제안하는 기법은 앞서 생성한 SCRS의 테스트 데이터를 포함하는 외부 입력들의 집합  $F$ 와 연관관계 집합  $R$ 을 이용하여, 집합  $R$ 의 구조에 따라 적응적으로 집합  $F$ 의 입력들을 조합함으로써 테스트 케이스를 생성한다.

### 2.1 모든 연관관계 조합 기법

연관관계 집합  $R$ 의 크기가 1이라면, 즉 집합  $R$ 이 모든 연관관계 구조라면, 축소된 테스트 케이스 집합( $T_{in}^{red}$ )을 생성하지 않고 식 (2)에서 보는 바와 같이 각 외부 입력 파라미터들의 테스트 데이터를 모두 조합(All combinations)하여 전체 테스트 케이스 집합( $T_m$ )을 생성하며, 이 때  $T_m$ 을 생성하기 위한 시간 복잡도는  $O(\prod_{f_i \in F} D(f_i))$ 이 된다.

### 2.2 독립된 연관관계 조합 기법

연관관계 집합  $R$ 의 연관관계들이 모두 독립된 연관관계라면 제안하는 Independent Union 기법을 통해 테스트 케이스를 생성한다. Independent Union 기법은 서로 독립적인 연관관계 집합들을 서로 합하여 축소된 테스트 케이스 집합( $T_{in}^{red}$ )을 생성하는 기법으로 기존의 Union 기법[15,16]과 달리 “don't care data”를 임의로 생성하여 합하지 않아 불필요한 테스트 케이스의 수를 줄인다.

그림 5는 Independent Union 기법에 대한 상세 동작 순서이다. 입력으로 테스트 데이터 집합을 포함하는 외부 입력 파라미터들의 집합  $F$ 와 독립적인 연관관계 집합  $R$ 를 가지며, 출력으로 테스트 케이스 집합  $T_{idk}$ 를 가진다. 본 기법은 먼저  $T_{idk}$ 를 초기화 하고 (line 1),  $R$ 의 각 연관관계  $r$ 마다 여기에 속한 외부 입력 파라미터들만의 테스트 데이터들을 서로 조합하여 테스트 케이스 집합을 생성한다(line 2-10). 좀 더 상세히 설명하면, 연관관계  $r$ 에 속한 외부 입력 파라미터들의 테스트 데이터 조합 수를 계산하여  $size$ 에 저장한다(line 3). 만약  $size$ 가 현재  $T_{idk}$ 의 크기( $|T_{idk}|$ )보다 크다면, ( $size - |T_{idk}|$ ) 개수만큼 “don't care data”를 포함하는 빈 테스트 케이스들을 생성하여  $T_{idk}$ 에 추가 한다(line 4-7). 그런 다음

```

Input : F: A set of external input parameters with test data
       R: A set of independent I/O relationships of SCBS
Output : T_idk : A set of test cases
Function IndependentUnion(F_idk, R_idk)
1  T_idk = ∅;
2  foreach r ∈ R do
3    size = ∏_{f_i ∈ r} D(f_i);
4    if size > |T_idk| then
5      // The empty test cases include "don't care data"
6      Add (size - |T_idk|) empty test cases into T_idk;
7    end
8    Fill all combinations of parameters in r into T_idk;
9  end
10 return T_idk;

```

그림 5. Independent Union 기법.

Fig. 5. Independent Union algorithm.

$r$ 에 속한 외부 입력 파라미터들의 테스트 데이터들을 모두 조합하여  $T_{idk}$ 에 채운다(line 8). 모든 연관관계에 대해 위와 같은 작업을 반복하여 최종적으로  $T_{idk}$ 를 출력한다(line 10). 그러므로 Independent Union 기법의 시간 복잡도는  $O(\sum_{k=1}^l (\prod_{f_i \in r_k} D(f_i)))$ 이며, 여기서  $l$ 는  $R$ 의 크기( $|R|$ )이다.

### 2.3 교집합 연관관계 조합 기법

연관관계 집합  $R$ 의 연관관계들이 교집합 연관관계라면 제안하는 HybridOrder 기법을 통해 테스트 케이스를 생성한다. HybridOrder 기법은 기존의 ParaOrder 기법[15]을 개선한 것으로서 집합  $R$ 에 속한 연관관계 집합들이 모두 독립적이지 않을 경우, 즉 각 연관관계 집합 간에 서로 교집합이 존재할 경우 축소된 테스트 케이스 집합( $T_{in}^{red}$ )을 생성하는 기법이다.

그림 6은 HybridOrder 기법에 대한 상세 동작 순서이다. 입력으로 테스트 데이터 집합을 포함하는 외부 입력 파라미터들의 집합  $F$ 와 연관관계 집합  $R$ 를 가지며, 출력으로 축소된 테스트 케이스 집합( $T_{in}^{red}$ )를 가진다. 본 기법은 독립된 연관관계별로 테스트 데이터를 조합( $T_{idk}$ ) 하는 단계(line 1-14), 독립된 연관관계에 포함되어 있지 않은 입력 파라미터별로 테스트 데이터를 조합( $T_{in}^{red}$ ) 하는 단계(line 15-26), 마지막으로  $T_{in}^{red}$ 에  $T_{idk}$ 를 합하는 단계(line 27-32)로 이루어져 있다.

각 단계별로 상세히 설명하면, 먼저  $R$ 에 속한 연관관계 중에서 독립된 연관관계 집합을 얻어  $R_{idk}$ 에 저장한다(line 2). 만약  $R_{idk}$ 가 공집합( $\emptyset$ )이 아니라면,  $R_{idk}$ 에 속한 외부 입

```

Input : F: A set of external input parameters with a set of test data
       R: A set of I/O relationships of SCBS
Output : T_in^red : A reduced set of test cases
1 //Combinations by independent relationships
2 R_idk = SelectInDepRelations(R);
3 if R_idk ≠ ∅ then
4   F_idk = (r_k ∩ F) | r_k ∈ R_idk;
5   T_idk = IndependentUnion(F_idk, R_idk);
6   F = F - F_idk;
7   R = R - R_idk;
8 end
9 r = SelectRelation(R);
10 size = ∏_{f_i ∈ r} D(f_i);
11 Create size empty test cases into T_idk;
12 Fill all combinations of parameters in r into T_idk;
13 F = F - {f_i | f_i ∈ r};
14 F_doad = {f_i | f_i ∈ r};
15 // Combinations by parameters
16 while F ≠ ∅ do
17   f_i = SelectFactor(F);
18   Inters = {r_k ∩ (F_doad + f_i) | f_i ∈ r_k, r_k ∈ R};
19   RedInters = Delete subset of elements in Inters;
20   CombSet = {(u_{i,1}, ..., u_{i,j}) | u_{i,1} ∈ V_{i,1}, ..., u_{i,j} ∈ V_{i,j}, {f_{i,1}, ..., f_{i,j}} ∈ RedInters};
21 // The follow extensions of T_idk^red are similar to ParaOrder
22 Extend T_idk^red horizontally;
23 Extend T_idk^red vertically;
24 F = F - {f_i};
25 F_doad = F_doad + {f_i};
26 end
27 // Union of T_idk and T_idk^red
28 if |T_idk| > |T_idk^red| then
29   // The empty test cases include "don't care data"
30   Add (|T_idk| - |T_idk^red|) empty test cases into T_idk^red;
31 end
32 Fill test cases in T_idk into T_idk^red;
33 return T_idk^red;

```

그림 6. HybridOrder 기법.

Fig. 6. HybridOrder algorithm.



력 파라미터들의 집합( $F_{id}$ )을 구해 Independent Union 기법을 통해 독립된 테스트 케이스 집합을 생성하여  $T_{id}$ 에 저장한다(line 4,5). 그런 다음 집합  $R$ 에서  $R_{id}$ 에 속한 연관관계 집합을 삭제하고, 집합  $F$ 에서  $F_{id}$ 에 속한 입력 파라미터들을 삭제한다(line 6,7). 독립된 연관관계 집합에 대한 테스트 케이스가 생성된 후 집합  $R$ 에서 크기가 가장 작은 연관관계  $r$ 를 선택하여( $SelectRelation()$ )  $r$ 에 속한 파라미터들의 테스트 데이터를 서로 조합하여  $T_{in}^{red}$ 에 추가한다(line 9-14). 이 때 집합  $F$ 에서  $r$ 에 속한 파라미터들을 삭제하고, 이미 생성한 입력 파라미터들의 집합  $F_{deal}$ 에  $r$ 에 속한 파라미터들을 추가한다. 그런 다음 집합  $F$ 에 속한 각 입력 파라미터에서 크기가 가장 작은  $f_i$ 를 선택하여( $SelectFactor()$ ),  $f_i$ 의 테스트 데이터를 ParaOrder 기법[17]과 유사한 방법으로  $T_{in}^{red}$ 에 추가한다(line 15-26). 이 때 기존의 ParaOrder 기법과는 달리  $f_i$ 를 포함하는 연관관계 집합  $r_k$ 와 ( $F_{deal}+f_i$ )과의 교집합의 합  $Inters=\{r_k \cap (F_{deal}+f_i) \mid f_i \in r_k, r_k \in R\}$ 에 부분집합이 존재 한다면 해당 부분집합을 삭제하여  $RedInters$ 를 생성하고(line 19),  $RedInters$ 의 각 원소에 속한 입력파라미터들의 테스트 데이터를 서로 조합하여  $CombSet$ 을 생성한다(line 20). 그리고 ParaOrder 기법과 유사하게  $CombSet$ ,  $D(f_i)$ 를 이용하여  $T_{in}^{red}$ 를 가로/세로로 확장한다(line 21-26). 마지막으로  $T_{id}$ 의 크기( $|T_{id}|$ )와  $T_{in}^{red}$ 의 크기( $|T_{in}^{red}|$ )를 비교하여,  $T_{id}$ 의 크기가 더 크다면 ( $|T_{id}| - |T_{in}^{red}|$ ) 만큼 빈 테스트 케이스를  $T_{in}^{red}$ 에 추가하여  $T_{id}$ 를 합 한다 (line 27-33).

HybridOrder 기법의 최악의 시간 복잡도는

$$O(\sum_{v=1}^w \prod_{f_q \in r_v} D(f_q) + \sum_{i=1}^z ((D(f_i) \times \sum_{k=1}^m \prod_{f_j \in r_k} D(f_j) \times |T_{id}|)))$$

이 되며,  $Inters$ 에 부분집합이 존재하지 않아 검색되는  $CombSet$ 의 크기가 줄어들며, 독립된 연관관계 집합이 존재할 시 불필요하게 각 외부 입력 파라미터마다 추가하지 않으므로 이러한 경우에는 단순해 질 수 있다.

#### IV. 실험 및 평가

본 장에서는 세 가지 실험을 통해 기존의 테스트 기법들 [16,17]과 비교하여 제안하는 기법의 효율성(effectiveness)을 검증하였다. 각 실험들의 목적은 SCRS의 입출력 연관관계 구조가 교집합 연관관계, 독립적 연관관계, 그리고 모든 연관관계 일 때 제안하는 기법이 축소된 테스트 케이스 집합을 생성하는지 검증하고, 각 입출력 연관관계 구조가 제안하는 기법에 어떠한 영향을 미치는지를 평가하는 것이다. 제안한 축소된 테스트 케이스 생성 기법의 효율성을 검증하기 위해 축소된 테스트 케이스를 생성 시 시간 복잡도를 계산하고, 다음과 같이 전체 테스트 케이스 축소율  $R_{tc}$ 를 계산하였다.

$$R_{tc}(\%) = \frac{|T_{in}| - |T_{in}^{red}|}{|T_{in}|} * 100$$

##### 1. 실험 대상

실험 1에서는 SCRS의 외부 입출력의 연관관계가 서로 교집합 관계를 가질 때 제안하는 기법들이 효율적으로 축소된 테스트 케이스 집합을 생성하는지를 검증한다.

표 1. 실험 1의 외부 입력 파라미터에 대한 명세 정보 및 테스트 데이터 집합.

Table 1. Test specifications and test data of input parameters in case study I.

외부 입력 파라미터( $x_i$ )	명세정보	테스트 데이터 ( $D(x_i)$ )
$x_{1,1}$	$x \leq 0$	{0,1}
$x_{1,2}$	$x \leq 0$	{0,1}
$x_{2,1}$	$x \leq 0$	{0,1}
$x_{2,2}$	$x \leq 0$	{0,1}
$x_{4,1}$	$x \leq 0$	{0,1}
$x_{4,2}$	$x \leq 0$	{0,1}
$x_{4,3}$	$x \leq 0$	{0,1}

##### 1.1 실험 1(교집합 연관관계)

실험 1에서 테스트 대상 SCRS는 그림 1의 예제 SCRS로서 4개의 단위 컴포넌트들로 구성되어 있다. 테스트 대상 SCRS의 외부 입력 파라미터 집합( $CC_{in}$ )과 외부 출력 파라미터 집합( $CC_{out}$ )은 제 2장에 따라 다음과 같이 정의된다.

$$CC_{in} = EXT_{in}(C_1) \cup EXT_{in}(C_2) \cup EXT_{in}(C_3) \cup EXT_{in}(C_4) = \{x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2}, x_{4,1}, x_{4,2}, x_{4,3}\},$$

$$CC_{out} = EXT_{out}(C_1) \cup EXT_{out}(C_2) \cup EXT_{out}(C_3) \cup EXT_{out}(C_4) = \{y_{1,1}, y_{1,2}, y_{3,1}, y_{3,2}, y_{4,1}\}$$

그리고 각 외부 입력 파라미터 마다 테스트 명세 정보를 이용하여 동등분할 기법[19]을 이용하여 테스트 데이터를 생성하였으며, 표 1은 SCRS의 각 외부 입력 파라미터별 테스트 명세 정보와 생성된 테스트 데이터 집합을 나타낸다.

그림 4의 IORG를 이용하여 테스트 데이터를 포함하는 외부 입력 파라미터 집합  $F$ 와 각 외부 입력 파라미터들간의 연관관계 집합  $R$ 을 생성하면 다음과 같다.

$F = \{f_1, f_2, f_3, f_4, f_5, f_6, f_7\}$ , 이때  $f_1 = \{x_{1,1}, D(x_{1,1})\}$ ,  $f_2 = \{x_{1,2}, D(x_{1,2})\}$ ,  $f_3 = \{x_{2,1}, D(x_{2,1})\}$ ,  $f_4 = \{x_{2,2}, D(x_{2,2})\}$ ,  $f_5 = \{x_{4,1}, D(x_{4,1})\}$ ,  $f_6 = \{x_{4,2}, D(x_{4,2})\}$ ,  $f_7 = \{x_{4,3}, D(x_{4,3})\}$ 이고,  $X(y_{1,1}) = \{x_{1,1}, x_{1,2}\}$ ,  $X(y_{1,2}) = \{x_{1,2}, x_{2,1}\}$ ,  $X(y_{3,1}) = \{x_{2,1}, x_{2,2}\}$ ,  $X(y_{3,2}) = \{x_{2,2}\}$ ,  $X(y_{4,1}) = \{x_{4,1}, x_{4,2}, x_{4,3}\}$ 이다. 이 때  $X(y_{3,2}) \subset X(y_{3,1})$ 이 되어  $X(y_{3,2})$ 의 연관관계는 삭제되어, 집합  $R$ 은 다음과 같이 생성된다.  $R = \{r_1, r_2, r_3, r_4\}$ 이며,  $r_1 = \{f_1, f_2\}$ ,  $r_2 = \{f_3, f_4\}$ ,  $r_3 = \{f_5, f_6, f_7\}$ 이 된다. 즉, 각 연관관계 집합  $r_1, r_2, r_3$ 는 교집합이 존재하며  $r_4$ 만 독립적인 관계가 된다.

그리고 전체 테스트 케이스 축소율  $R_{tc}$ 를 측정하고자 먼저 각 외부 입력 파라미터의 테스트 데이터를 모두 조합(All combination)하여 다음과 같이 128개로 구성된 전체 테스트 케이스 집합을 생성하였다.

$$T_{in} = D(x_{1,1}) \times D(x_{1,2}) \times D(x_{2,1}) \times D(x_{2,2}) \times D(x_{4,1}) \times D(x_{4,2}) \times D(x_{4,3})$$

$$|T_{in}| = 2 * 2 * 2 * 2 * 2 * 2 * 2 = 128.$$

##### 1.2 실험 2(독립적 연관관계)

실험 2에서 테스트 대상 SCRS는 장애물을 피해 로봇을 이동시키면서 안내 메시지를 출력하는 시스템으로서 그림 7(a)에서 보는 바와 같이 4개의 단위 컴포넌트들, 거리 센

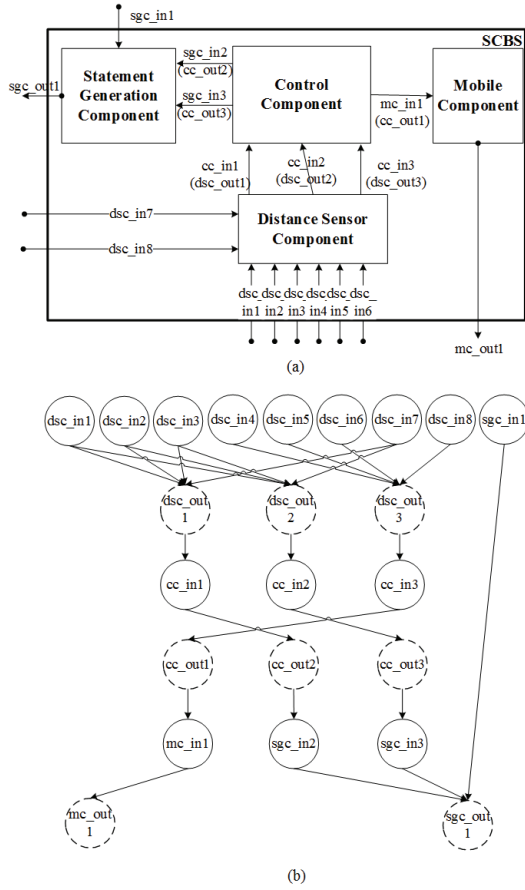


그림 7. 실험 2의 테스트 대상 SCRS 구조.

Fig. 7. The SCRS architecture in case study II.

서 컴포넌트(Distance sensor component,  $C_{dsc}$ ), 모바일 컴포넌트(Mobile component,  $C_{mc}$ ), 문장 생성 컴포넌트(Statement generation component,  $C_{sgc}$ ), 제어 컴포넌트(Control component,  $C_{cc}$ )로 구성 되어 있다. 테스트 대상 SCRS의 외부 입력 파라미터 집합( $CC_{in}$ )과 외부 출력 파라미터 집합( $CC_{out}$ )은 다음과 같이 정의된다.

$$\begin{aligned} CC_{in} &= EXT_{in}(C_{dsc}) \cup EXT_{in}(C_{mc}) \cup EXT_{in}(C_{sgc}) \cup EXT_{in}(C_{cc}) \\ &= \{dsc\_in1, dsc\_in2, dsc\_in3, dsc\_in4, dsc\_in5, \\ &\quad dsc\_in6, dsc\_in7, dsc\_in8, sgc\_in1\}, \end{aligned}$$

$$\begin{aligned} CC_{out} &= EXT_{out}(C_{dsc}) \cup EXT_{out}(C_{mc}) \cup EXT_{out}(C_{sgc}) \cup EXT_{out}(C_{cc}) \\ &= \{mc\_out1, sgc\_out1\}. \end{aligned}$$

그리고 테스트 명세 정보를 이용하여 각 외부 입력 파라미터마다 제한한 확장된 명세기반 테스트 데이터 생성 기법을 통해 테스트 데이터를 생성하였으며, 표 2는 SCRS의 각 외부 입력 파라미터별 테스트 명세 정보와 생성된 테스트 데이터 집합을 나타낸다.

그림 7(b)는 테스트 대상 SCRS에 대한 실제 입출력 연관관계를 기반으로 생성한 IORG를 나타낸다. 실제 IORG와  $T_d$ 를 이용하여 테스트 데이터를 포함하는 외부 입력 파라미터 집합  $F$ 와 각 외부 입력 파라미터들간의 연관관계 집합  $R$ 을 생성하면 다음과 같다.

표 2. 실험 2의 외부 입력 파라미터에 대한 명세 정보 및 테스트 데이터 집합.

Table 2. Test specifications and test data of input parameters in case study II.

외부 입력 파라미터 ( $x_i$ )	명세정보	테스트 데이터 ( $D(x_i)$ )
$dsc\_in1$	$0 < x \leq 1, 1 < x \leq 2$ $2 < x \leq 3, 3 < x$	{0.5, 1.2, 2.9, 5.0}
$dsc\_in2$	$0 < x \leq 1, 1 < x \leq 2$ $2 < x \leq 3, 3 < x$	{0.5, 1.2, 2.2, 3.1}
$dsc\_in3$	$0 < x \leq 1, 1 < x \leq 2$ $2 < x \leq 3, 3 < x$	{0.5, 1.2, 2.2, 4.0}
$dsc\_in4$	$0 < x \leq 1, 1 < x \leq 2$ $2 < x$	{1.0, 1.2, 2.8}
$dsc\_in5$	$0 < x \leq 1, 1 < x \leq 2$ $2 < x$	{0.3, 1.2, 3.4}
$dsc\_in6$	$0 < x \leq 1, 1 < x \leq 2$ $2 < x$	{0.9, 2.0, 4.2}
$dsc\_in7$	$0 < x \leq 3, 3 < x$	{2.0, 3.1}
$dsc\_in8$	$0 < x \leq 2, 2 < x$	{1.2, 2.8}
$sgc\_in1$	Korea, English	{KOREA, ENGLISH}

$F = \{f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9\}$ , 이때  $f_1 = \{dsc\_in1, D(dsc\_in1)\}$ ,  $f_2 = \{dsc\_in2, D(dsc\_in2)\}$ ,  $f_3 = \{dsc\_in3, D(dsc\_in3)\}$ ,  $f_4 = \{dsc\_in4, D(dsc\_in4)\}$ ,  $f_5 = \{dsc\_in5, D(dsc\_in5)\}$ ,  $f_6 = \{dsc\_in6, D(dsc\_in6)\}$ ,  $f_7 = \{dsc\_in7, D(dsc\_in7)\}$ ,  $f_8 = \{dsc\_in8, D(dsc\_in8)\}$ ,  $f_9 = \{sgc\_in1, D(dsc\_in1)\}$  이고,  $X(sgc\_out1) = \{dsc\_in1, dsc\_in2, dsc\_in3, dsc\_in7, sgc\_in1\}$ ,  $X(mc\_out1) = \{dsc\_in4, dsc\_in6, dsc\_in6, dsc\_in8\}$  이므로,  $R = \{r_1, r_2\}$ , 이때  $r_1 = \{f_1, f_2, f_3, f_7, f_9\}$ ,  $r_2 = \{f_4, f_5, f_6, f_8\}$ . 즉, 각 외부 입력 파라미터들간의 연관관계  $r_1 \cap r_2 = \emptyset$ 이므로 서로 독립된 연관관계를 가진다.

그리고 전체 테스트 케이스 축소율  $R_c$ 를 측정하고자 먼저 각 외부 입력 파라미터의 테스트 데이터를 모두 조합(All combination)하여 다음과 같이 13824개로 구성된 전체 테스트 케이스 집합을 생성하였다.

$$\begin{aligned} T_{in} &= D(dsc\_in1) \times D(dsc\_in2) \times D(dsc\_in3) \\ &\quad \times D(dsc\_in4) \times D(dsc\_in5) \times D(dsc\_in6) \\ &\quad \times D(dsc\_in7) \times D(dsc\_in8) \times D(sgc\_in1), \\ |T_{in}| &= 4*4*4*3*3*3*2*2 = 13824. \end{aligned}$$

### 1.3 실험 3(모든 연관관계)

실험 3에서 테스트 대상 SCRS는 네비게이션(Navigation) 기능을 이용하여 로봇을 이동시키면서 안내 메시지를 출력하는 시스템으로서 그림 8(a)에서 보는 바와 같이 10개의 단위 컴포넌트들, 전역 맵 컴포넌트(Global map component,  $C_{gmc}$ ), 경로 생성 컴포넌트(Path planning component,  $C_{ppc}$ ), 장애물 회피 컴포넌트(Obstacle avoidance component,  $C_{oac}$ ), 네비게이션 컴포넌트(Navigation component,  $C_{nc}$ ), 위치인식 컴포넌트(Localization component,  $C_{lc}$ ), 범퍼 센서 컴포넌트(Bumper sensor component,  $C_{bsc}$ ), 사용자 감지 컴포넌트(User detection component,  $C_{udc}$ ), 레이저 스캐너 컴포넌트(Laser scanner component,  $C_{isc}$ ), 모바일 컴포넌트(Mobile component,

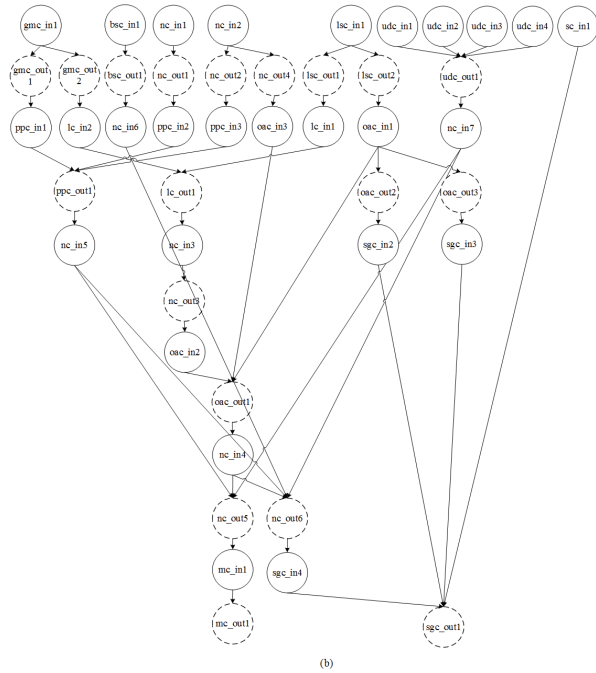
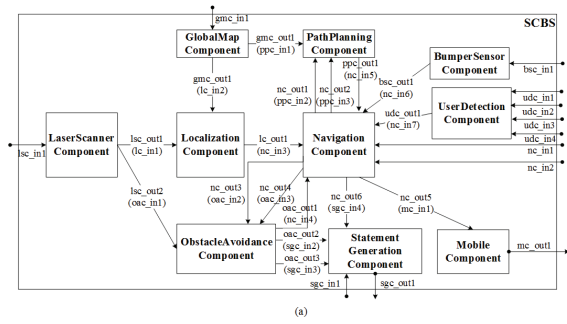


그림 8. 실험 3의 테스트 대상 SCRS 구조.

Fig. 8. The SCRS architecture in case study III.

$C_{mc}$ ), 문장 생성 컴포넌트(Statement generation component,  $C_{sgc}$ )로 구성되어 있다. 테스트 대상 SCRS의 외부 입력 파라미터 집합( $CC_{in}$ )과 외부 출력 파라미터 집합( $CC_{out}$ )은 제 2장에 따라 다음과 같이 정의된다.

$$\begin{aligned}
 CC_{in} &= EXT_{in}(C_{gmc}) \cup EXT_{in}(C_{ppc}) \cup EXT_{in}(C_{lsc}) \cup EXT_{in}(C_{lc}) \\
 &\quad \cup EXT_{in}(C_{nc}) \cup EXT_{in}(C_{bsc}) \cup EXT_{in}(C_{udc}) \cup EXT_{in}(C_{oac}) \\
 &\quad \cup EXT_{in}(C_{sgc}) \cup EXT_{in}(C_{mc}) \\
 &= \{gmc\_in1, lsc\_in1, nc\_in1, nc\_in2, bsc\_in1, \\
 &\quad udc\_in1, udc\_in2, udc\_in3, udc\_in4, sgc\_in1\}, \\
 CC_{out} &= EXT_{out}(C_{gmc}) \cup EXT_{out}(C_{ppc}) \cup EXT_{out}(C_{lsc}) \cup EXT_{out}(C_{lc}) \\
 &\quad \cup EXT_{out}(C_{nc}) \cup EXT_{out}(C_{bsc}) \cup EXT_{out}(C_{udc}) \cup EXT_{out}(C_{oac}) \\
 &\quad \cup EXT_{out}(C_{sgc}) \cup EXT_{out}(C_{mc}) \\
 &= \{sgc\_out1, mc\_out1\}.
 \end{aligned}$$

그리고 테스트 명세 정보를 이용하여 각 외부 입력 파라미터마다 제한한 확장된 명세기반 테스트 데이터 생성 기법을 통해 테스트 데이터를 생성하였으며, 표 3은 SCRS의 각 외부 입력 파라미터별 테스트 명세 정보와 생성된 테스트 데이터 집합을 나타낸다.

표 3. 실험 3의 외부 입력 파라미터에 대한 명세 정보 및 테스트 데이터 집합.

Table 3. Test specifications and test data of input parameters in case study III.

외부 입력 파라미터 ( $x_i$ )	명세정보	테스트 데이터 ( $D(x_i)$ )
$gmc\_in1$	valid map file	{officeMap.map}
$lsc\_in1$	source sensor, destination sensor, intermediate sensor, left obstacle sensor, right obstacle sensor, middle obstacle sensor,	{SRC, DEST, INTER, LEFT, RIGHT, MIDDLE}
$nc\_in1$	valid source coordinate, invalid source coordinate	{VALID-SRC, INVALID-SRC}
$nc\_in2$	valid destination coordinate, invalid destination coordinate	{VALID-DEST, INVALID-DEST}
$bsc\_in1$	bumper is not touched, bumper is touched	{0, 1}
$udc\_in1$	$0 < x \leq 1, 1 < x \leq 2, 2 < x \leq 3, 3 < x$	{0.5, 1.2, 2.9, 5.0}
$udc\_in2$	$0 < x \leq 1, 1 < x \leq 2, 2 < x \leq 3, 3 < x$	{0.5, 1.2, 2.2, 3.1}
$udc\_in3$	$0 < x \leq 1, 1 < x \leq 2, 2 < x \leq 3, 3 < x$	{0.5, 1.2, 2.2, 4.0}
$udc\_in4$	$0 < x \leq 3, 3 < x$	{2.0, 3.1}
$sgc\_in1$	Korea, English	{KOREA, ENGLISH}

그림 8(b)는 테스트 대상 SCRS에 대한 실제 입출력 연관관계를 기반으로 생성한 IORG를 나타낸다. 실제 IORG와  $T_d$ 를 이용하여 테스트 데이터를 포함하는 외부 입력 파라미터 집합  $F$ 와 각 외부 입력 파라미터들간의 연관관계 집합  $R$ 을 생성하면 다음과 같다.

$F = \{f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9, f_{10}\}$ , 이때  $f_i = \{gmc\_in1, D(gmc\_in1)\}$ ,  $f_2 = \{lsc\_in1, D(lsc\_in1)\}$ ,  $f_3 = \{nc\_in1, D(nc\_in1)\}$ ,  $f_4 = \{nc\_in2, D(nc\_in2)\}$ ,  $f_5 = \{bsc\_in1, D(bsc\_in1)\}$ ,  $f_6 = \{udc\_in1, D(udc\_in1)\}$ ,  $f_7 = \{udc\_in2, D(udc\_in2)\}$ ,  $f_8 = \{udc\_in3, D(udc\_in3)\}$ ,  $f_9 = \{udc\_in4, D(udc\_in4)\}$ ,  $f_{10} = \{sgc\_in1, D(sgc\_in1)\}$  이고,  $X(sgc\_out1) = CC_{in}$ ,  $X(mc\_out1) = \{gmc\_in1, lsc\_in1, nc\_in1, nc\_in2, bsc\_in1, udc\_in1, udc\_in2, udc\_in3, udc\_in4\}$  이고,  $X(sgc\_out1) \supset X(mc\_out1)$ 의 관계를 가지므로  $R = \{r_i\}$ , 이때  $r_i = \{f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9, f_{10}\}$  이 된다. 즉, SCRS의 특정 외부 출력이 모든 외부 입력에 영향을 받으므로 축소된 테스트 케이스 집합은 전체 테스트 케이스 집합과 동일하게 된다. 전체 테스트 케이스의 수는 다음과 같이 계산될 수 있다.

$$\begin{aligned}
 T_m &= D(gmc\_in1) \times D(lsc\_in1) \times D(nc\_in1) \times D(nc\_in2) \\
 &\quad \times D(bsc\_in1) \times D(udc\_in1) \times D(udc\_in2) \times D(udc\_in3) \\
 &\quad \times D(udc\_in4) \times D(sgc\_in1), \\
 |T_m| &= 1 * 6 * 2 * 2 * 2 * 4 * 4 * 4 * 2 * 2 = 12288.
 \end{aligned}$$

표 4. 실험들에 대한 전체 테스트 케이스 축소율 및 시간 복잡도.

Table 4. The time complexity and rate of reduction of the complete test cases in case studies.

실험	전체 테스트 케이스 수 ( $T_m$ )	생성기법	테스트 케이스 축소율		시간 복잡도	
			$R_c$	$ T_m^{red} $	$O(\cdot)$	비 고
1 (교집합)	128	Union [16]	84.37%	20	20	$O(\sum_{k=1}^4 \prod_{f_i \in r_k} D(f_i))$
		ParaOrder [17]	93.75%	8	218	$O(\sum_{i=1}^7 ((D(f_i) \times \sum_{k=1}^m \prod_{f_j \in r_k} D(f_j) \times  T_{ci} )))$
		제안한 기법	93.75%	8	76	$O(\prod_{f_i \in r_1} D(f_i) + \sum_{i=1}^4 ((D(f_i) \times \sum_{k=1}^m \prod_{f_j \in r_k} D(f_j) \times  T_{ci} )))$
2 (독립적)	13824	Union [16]	97.63%	310	310	$O(\sum_{k=1}^2 \prod_{f_i \in r_k} D(f_i))$
		ParaOrder [17]	98.02%	256	86154	$O(\sum_{i=1}^9 ((D(f_i) \times \sum_{k=1}^m \prod_{f_j \in r_k} D(f_j) \times  T_{ci} )))$
		제안한 기법	98.02%	256	310	$O(\sum_{k=1}^2 (\prod_{f_i \in r_k} D(f_i)))$
3 (모든조합)	12288	Union [16]	0.0%	12288	12288	$O(\sum_{k=1}^1 \prod_{f_i \in r_k} D(f_i)) = O(\prod_{f_i \in F} D(f_i))$
		ParaOrder [17]	0.0%	12288	100663296	$O(\sum_{i=1}^{10} ((D(f_i) \times \sum_{k=1}^m \prod_{f_j \in r_k} D(f_j) \times  T_{ci} )))$
		제안한 기법	0.0%	12288	12288	$O(\prod_{f_i \in F} D(f_i))$

2. 실험 결과

본 논문에서 제안한 기법의 효율성을 평가하고자 표 4에서 보는 바와 같이 각 실험마다 전체 테스트 케이스 축소율  $R_c$ 와 시간 복잡도를 측정하였다.

교집합 연관관계 구조를 가지는 실험 1의 테스트 케이스 축소율 결과, 128개로 구성된 전체 테스트 케이스 집합에 대해, Union 기법[16]은 20개로 구성된 축소된 테스트 케이스 집합을 생성하여 84.37% 테스트 케이스 수를 감소시켰으며, 제안한 기법과 ParaOrder 기법[17]은 8개의 축소된 테스트케이스를 생성하여 93.75% 테스트 케이스 수를 감소시켰다. Union 기법의 경우  $r_1, r_2, r_3$ , 각 연관관계에 포함된 외부 입력 파라미터에 대한 테스트 데이터를 서로 조합하여 각 4개씩 테스트 케이스를 생성하고,  $r_4$ 에 대해 8개의 테스트 케이스를 생성한다. 이 때 각 연관관계에 포함되지 않은 외부 입력 파라미터에 대해서는 미리 랜덤하게 테스트 데이터를 추가한 후 합하여 20개의 축소된 테스트 케이스를 생성하였다. 제안한 기법과 ParaOrder 기법은 축소된 테스트 케이스 생성 시 각 연관관계에 포함되지 않은 외부 입력 파라미터에 대해 미리 처리를 하지 않고 각 테스트 데이터를 조합하기 때문에 8개의 축소된 테스트 케이스를 생성한다. 그리고 축소된 테스트 케이스 생성을 위한 시간 복잡도 계산 결과, Union 기법의 시간 복잡도는 연관관계 집합 개수에 따라 계산되어 단순하며, ParaOrder 기법은 각 외부 입력 파라미터들을 하나씩 추가함으로써 축소된 테스트 케이스를 생성하므로 시간 복잡도가 증가한다. 제안한 기법은 연관관계 중  $r_1, r_2, r_3$ 와 교집합이 존재하지 않는  $r_4$ 에 대한 조합을 먼저 생성하고 나머지 외부 입력 파라미터들을 하나씩 조합함에 따라 ParaOrder 기법에 비해 시간 복잡도가 간소화됨을 확인 할 수 있었다.

독립된 연관관계 구조를 가지는 실험 2의 테스트 케이스 축소율 결과, 13824개로 구성된 전체 테스트 케이스 집합에 대해, Union 기법[16]은 310개 구성된 축소된 테스트 케이스 집합을 생성하여 97.63% 테스트 케이스 수를 감소시켰으며, 제안한 기법과 ParaOrder 기법[17]은 256개의 축소된 테스트 케이스를 생성하여 98.02% 테스트 케이스 수를 감소시켰다. Union 기법의 경우 연관관계  $r_1$ 에 포함된 각 외부 입력 파라미터에 대한 테스트 데이터를 서로 조합하여 256개의 테스트 케이스를 생성하고 연관관계  $r_2$ 에 포함된 각 외부 입력 파라미터들을 서로 조합하여 54개의 테스트 케이스를 생성한다. 이 때 각 연관관계에 포함되지 않은 외부 입력 파라미터에 대해서는 미리 랜덤하게 테스트 데이터를 추가한 후 합하여 310개의 축소된 테스트 케이스를 생성한다. 제안한 기법과 ParaOrder 기법은 축소된 테스트 케이스 생성 시 각 연관관계에 포함되지 않은 외부 입력 파라미터에 대해 미리 처리를 하지 않고 각 테스트 데이터를 조합하기 때문에 256개의 축소된 테스트 케이스를 생성한다. 그리고 축소된 테스트 케이스 생성을 위한 시간 복잡도 계산 결과, Union 기법과 제안한 기법은 연관관계 집합 개수에 따라 시간 복잡도가 계산되어 동일한 시간 복잡도를 가지며, ParaOrder 기법은 각 외부 입력 파라미터 개수에 따라 시간 복잡도가 계산되어 제안한 기법과 동일한 테스트 케이스 감소율을 보이지만 시간 복잡도는 증가함을 확인 할 수 있었다.

모든 연관관계 구조를 가지는 실험 3의 테스트 케이스 축소율 결과, 12288개로 구성된 전체 테스트 케이스 집합에 대해, Union [16], ParaOrder [17], 제안한 기법 모두 축소율은 0%이며, 그 원인은 외부 입력 파라미터들간의 연관관계 집합  $R$ 에 속한 연관관계가 1개( $r_1$ )이기 때문이다. 이럴 경우 각 기법은 전체 테스트 케이스 집합과 동일한 테스트 케이



스를 생성하게 되지만 시간 복잡도는 다름을 확인할 수 있었다. Union 기법과 제안한 기법은 한번에 외부 입력 파라미터의 테스트 데이터를 조합하여 단순한 시간 복잡도를 가지지만, ParaOrder 기법은 각 외부 입력 파라미터를 하나씩 추가하면서 테스트 데이터를 조합하며, 매번 기존에 조합된 테스트 케이스를 검색함에 따라 시간 복잡도가 증가함을 확인하였다.

각 실험들의 결과를 통해 제안한 기법이 SCBS의 연관관계 구조가 다를지라도 Union 기법[16]에 비해 테스트 케이스 축소율이 우수하며, ParaOrder 기법[17]에 비해 시간 복잡도가 우수함을 증명하였다.

## V. 결론

본 논문에서는 소프트웨어 컴포넌트 기반 로봇 시스템 테스트 시 소비되는 시간 및 비용을 감소시키고자 테스트 케이스를 자동으로 생성하는 기법을 제안하였다.

제안하는 기법은 테스트 관점에서 소프트웨어 컴포넌트 기반 로봇 시스템을 정의하고, 세 가지 타입의 시스템 입출력 연관관계 구조, 즉 독립적 연관관계, 모든 연관관계, 교집합 연관관계를 정의하였다. 그리고 시스템의 입출력 연관관계 정보를 기반으로 시스템의 입력 연관관계 집합을 생성하고, 시스템의 입출력 연관관계 구조에 따라 각 입력 파라미터의 테스트 데이터들을 적응적(adaptive)으로 조합하여 테스트 케이스를 생성하였다. 제안한 기법의 효율성을 검증하기 위해 세 가지 타입의 입출력 연관관계 구조를 가진 시스템들에 대해 전체 테스트 케이스 축소율과 시간 복잡도를 계산하여 제안한 기법의 우수함을 보였다.

향후 연구 과제로 비 결정론적(non-deterministic) 로봇 시스템에 대한 테스트 케이스와 예상 결과 값을 자동으로 생성하기 위한 연구를 진행할 것이다.

## REFERENCES

- [1] A. Brooks, T. Kaupp, A. Makarenko, S. Williams, and A. Oreback, "Towards component-based robotics," in: *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005, (IROS 2005)*, pp. 163-168, 2005.
- [2] D. Brugali and P. Scandurra, "Component-based robotic engineering (part 1) [tutorial]," *IEEE Robot. Autom. Magaz.*, vol. 16, no. 4, pp. 84-96, 2009.
- [3] S. Han, M. Sook Kim, and H. S. Park, "Open software platform for robotic services," *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 3, pp. 467-481, 2012.
- [4] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source robot operating system," in: *ICRA Workshop on Open Source Software*, vol. 3, 2009.
- [5] N. Ando, T. Suehiro, K. Kitagaki, T. Kotoku, and W.-K. Yoon, "RT-middleware: distributed component middleware for RT (robot technology)," in: *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2005 (IROS 2005)*, pp. 3933-3938, 2005.
- [6] C. S. Jang, B. Y. Song, and S. H. Kim, "Event port extension of OPRoS framework for inter-connecting with ROS topic," *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 20, no. 12, pp. 1252-1258, Dec. 2014.
- [7] G. J. Myers, "The Art of Software Testing," John Will & Sons, 1979.
- [8] J. S. Kang, H. S. Choi, S. W. Maeng, S. W. Kim, and H. S. Park, "SITAT: Simulation-based Interface testing automation tool for robot software component," *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 16, no. 6, pp. 608-616, Jun. 2010.
- [9] S. W. Maeng and H. S. Park, "Generation Algorithm of test suite for state transition sequence with abnormal transitions in robot software component," *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 16, no. 8, pp. 786-793, Aug. 2010.
- [10] S. H. Kim and H. S. Park, "Design of a robot-in-the-loop simulation based on OPRoS," *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 19, no. 3, pp. 248-255, Mar. 2013.
- [11] D. M. Cohen, S. R. Dalal, and M. L. Fredman, et al., "The AETG system: An approach to testing based on combinatorial design," *IEEE Trans. on Software Engineering*, vol. 23, no. 7, pp. 437-444, 1997.
- [12] K. C. Tai and Y. Lei, "A test generation strategy for pairwise testing," *IEEE Trans. on Software Engineering*, vol. 28, no. 1, pp. 109-111, 2002.
- [13] C. J. Colbourn, M. B. Cohen, and R. C. Turban, "A deterministic density algorithm for pairwise interaction coverage," in: *Proceedings of IASTED International Conference on Software Engineering (SE2004)*, Innsbruck, Austria, pp. 345-352, 2004.
- [14] N. Kobayashi, T. Tsuchiya, and T. Kikuno, "A new method for constructing pair-wise covering designs for software testing," *Information Processing Letters*, vol. 81, no. 2, pp. 85-91, 2002.
- [15] P. J. Schroeder and B. Korel, "Black-box test reduction using input-output analysis," in: *Proc. of the International Symposium on Software Testing and Analysis (ISSTA'00)*, Portland, Oregon, pp. 21-22, Aug. 2000.
- [16] P. J. Schoeder, "Black-box test reduction using input-output analysis," Ph.D. Thesis, Department of Computer Science, Illinois Institute of Technology, Chicago, IL, USA, 2001.
- [17] W. Ziyuan, N. Changhai, and X. Baowen, "Generating combinatorial test suite for interaction relationship," *SOQUA '07 Fourth International Workshop on Software Quality Assurance: in Conjunction with the 6th*

*ESEC/FSE Joint Meeting, 2007.*

- [18] J. S. Kang and H. S. Park, "Automatic generation algorithm of expected results for testing of component-based software system," *Information and Software Technology*, vol. 57, pp. 1-20, Jan. 2015.
- [19] T. J. Ostrand and M. J. Balcer, "The category-partition method for specifying and generating functional tests," *Commun. ACM*, vol. 31, no. 6, pp. 676-686, 1988.
- [20] D. Hoffman, P. Strooper, and L. White, "Boundary values and automated component testing," *Softw. Test., Verif. Reliab.*, vol. 9, no. 1, pp. 3-26, 1999.



### 강정석

2006년 강원대학교 전자통신공학과 졸업. 2008년 동 대학원 석사. 2015년 동 대학원 박사. 관심분야는 소프트웨어 테스트 자동화, 개발 프로세스 개선, 통신 미들웨어.



### 박홍성

1983년 서울대학교 제어계측공학과 졸업. 1986년 동 대학원 석사. 1992년 동 대학원 박사. 1992년~현재 강원대학교 전자통신공학과 교수. 관심분야는 로봇 S/W 플랫폼, 무선데이터통신.