



**(19) 대한민국특허청(KR)**  
**(12) 등록특허공보(B1)**

(45) 공고일자 2015년12월30일  
 (11) 등록번호 10-1581615  
 (24) 등록일자 2015년12월23일

(51) 국제특허분류(Int. Cl.) G06F 11/36 (2006.01) (52) CPC특허분류 G06F 11/3624 (2013.01) (21) 출원번호 10-2015-0010621 (22) 출원일자 2015년01월22일 심사청구일자 2015년01월22일 (56) 선행기술조사문헌 KR101419646 B1* ‘Automatic Generation Algorithm of Expected Results for Testing of Component-Based Software System’, Jeong Seok Kang, Hong Seong Park(2014)* *는 심사관에 의하여 인용된 문헌	(73) 특허권자 강원대학교산학협력단 강원도 춘천시 강원대학길 1 (효자동) (72) 발명자 박홍성 서울특별시 광진구 광나루로56길 32, 212동 2004호 (구의3동, 현대아파트2단지) 강정석 전라북도 고창군 아산면 갑평1길 39-1 (74) 대리인 특허법인 무한
---	--

전체 청구항 수 : 총 17 항

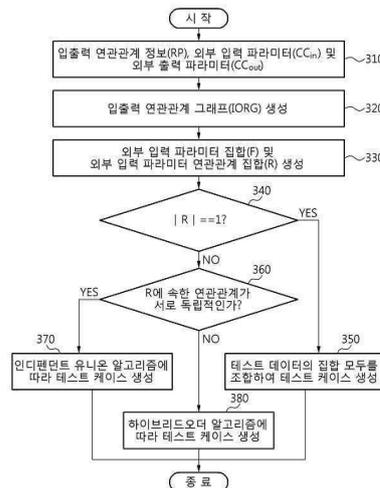
심사관 : 이동하

(54) 발명의 명칭 **소프트웨어 컴포넌트 기반 시스템을 위한 입출력 연관관계 기반 적응형 조합 테스트 장치 및 방법**

**(57) 요약**

소프트웨어 컴포넌트 기반 시스템을 위한 테스트 장치 및 방법이 개시된다. 소프트웨어 컴포넌트 기반 시스템을 위한 테스트 장치는 소프트웨어 컴포넌트 기반 시스템의 테스트 데이터의 집합을 포함하는 외부 입력 파라미터 집합 및 외부 입력 파라미터 연관관계 집합을 생성하고, 상기 외부 입력 파라미터 연관관계 집합에 기초하여 상기 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 구조의 유형을 결정하고, 상기 결정된 입출력 연관관계 구조의 유형에 따라 상이한 알고리즘에 의해 테스트 데이터 조합을 생성할 수 있다.

**대표도** - 도3



이 발명을 지원한 국가연구개발사업

과제고유번호 10044006

부처명 산업통상자원부

연구관리전문기관 산업기술평가관리원

연구사업명 산업원천기술개발사업

기술 개발 연구과제명 사용자 편의 중심의 개발도구와 로봇 표준 API 컴포넌트를 지원하는 개방형 로봇 미들웨어

기 여 율 1/1

주관기관 한국전자통신연구원

연구기간 2012.12.01 ~ 2017.11.30

---

## 명세서

### 청구범위

#### 청구항 1

소프트웨어 컴포넌트 기반 시스템을 위한 테스트 장치에 있어서,

소프트웨어 컴포넌트 기반 시스템의 테스트 데이터의 집합을 포함하는 외부 입력 파라미터 집합 및 외부 입력 파라미터 연관관계 집합을 생성하고, 상기 외부 입력 파라미터 연관관계 집합에 기초하여 상기 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 구조의 유형을 결정하고, 상기 결정된 입출력 연관관계 구조의 유형에 따라 상이한 알고리즘에 의해 테스트 데이터 조합을 생성하는 프로세서

를 포함하고,

상기 입출력 연관관계 구조의 유형은, 일부의 외부 출력에 대하여 모든 외부 입력이 영향을 주는 연관관계 구조를 나타내는 제1 유형, 각각의 외부 출력에 대하여 서로 독립적인 외부 입력이 영향을 주는 연관관계 구조를 나타내는 제2 유형 및 각각의 외부 출력에 대하여 서로 교집합을 가지는 외부 입력이 영향을 주는 연관관계 구조를 나타내는 제3 유형을 포함하는, 소프트웨어 컴포넌트 기반 시스템을 위한 테스트 장치.

#### 청구항 2

제1항에 있어서,

상기 소프트웨어 컴포넌트 기반 시스템의 외부 입력 파라미터, 상기 소프트웨어 컴포넌트 기반 시스템의 외부 출력 파라미터 및 상기 소프트웨어 컴포넌트 기반 시스템 내의 단위 컴포넌트 간의 입출력 연관관계 정보 및 상기 단위 컴포넌트 내부의 입출력 연관관계 정보를 입력받는 입력부

를 더 포함하는, 소프트웨어 컴포넌트 기반 시스템을 위한 테스트 장치.

#### 청구항 3

제1항에 있어서,

상기 프로세서는 상기 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 그래프를 생성하고, 상기 입출력 연관관계 그래프를 이용하여 상기 외부 입력 파라미터 연관관계 집합을 생성하는,

소프트웨어 컴포넌트 기반 시스템을 위한 테스트 장치.

#### 청구항 4

제3항에 있어서,

상기 프로세서는 상기 소프트웨어 컴포넌트 기반 시스템 내의 단위 컴포넌트 간의 입출력 연관관계 정보 및 상기 단위 컴포넌트 내부의 입출력 연관관계 정보에 기초하여 상기 입출력 연관관계 그래프를 생성하는,

소프트웨어 컴포넌트 기반 시스템을 위한 테스트 장치.

#### 청구항 5

삭제

#### 청구항 6

제1항에 있어서,

상기 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 구조가 제1 유형으로 결정된 경우, 상기 프로세서는 상기 테스트 데이터의 집합 모두를 조합하여 상기 테스트 데이터 조합을 생성하는,

소프트웨어 컴포넌트 기반 시스템을 위한 테스트 장치.

**청구항 7**

제1항에 있어서,

상기 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 구조가 제2 유형으로 결정된 경우, 상기 프로세서는 인디펜던트 유니온(Independent Union) 알고리즘에 따라 상기 외부 입력 파라미터 집합 및 상기 외부 입력 파라미터 연관관계 집합에 기초하여 상기 테스트 데이터 조합을 생성하는,

소프트웨어 컴포넌트 기반 시스템을 위한 테스트 장치.

**청구항 8**

제1항에 있어서,

상기 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 구조가 제3 유형으로 결정된 경우, 상기 프로세서는 하이브리드오더(HybridOrder) 알고리즘에 따라 상기 외부 입력 파라미터 집합 및 상기 외부 입력 파라미터 연관관계 집합에 기초하여 상기 테스트 데이터 조합을 생성하는,

소프트웨어 컴포넌트 기반 시스템을 위한 테스트 장치.

**청구항 9**

소프트웨어 컴포넌트 기반 시스템을 위한 테스트 장치에 있어서,

소프트웨어 컴포넌트 기반 시스템 내의 단위 컴포넌트 간의 입출력 연관관계 정보에 기초하여 상기 입출력 연관관계 그래프를 생성하고, 상기 소프트웨어 컴포넌트 기반 시스템의 테스트 데이터의 집합을 포함하는 외부 입력 파라미터 집합 및 외부 입력 파라미터 연관관계 집합을 생성하고, 상기 외부 입력 파라미터 연관관계 집합에 기초하여 상기 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 구조의 유형을 결정하고, 상기 결정된 입출력 연관관계 구조의 유형에 따라 상이한 알고리즘에 의해 테스트 데이터 조합을 생성하는 프로세서를

포함하고,

상기 입출력 연관관계 구조의 유형은, 일부의 외부 출력에 대하여 모든 외부 입력이 영향을 주는 연관관계 구조를 나타내는 제1 유형, 각각의 외부 출력에 대하여 서로 독립적인 외부 입력이 영향을 주는 연관관계 구조를 나타내는 제2 유형 및 각각의 외부 출력에 대하여 서로 교집합을 가지는 외부 입력이 영향을 주는 연관관계 구조를 나타내는 제3 유형을 포함하는, 소프트웨어 컴포넌트 기반 시스템을 위한 테스트 장치.

**청구항 10**

제9항에 있어서,

상기 소프트웨어 컴포넌트 기반 시스템의 외부 입력 파라미터, 상기 소프트웨어 컴포넌트 기반 시스템의 외부 출력 파라미터 및 상기 소프트웨어 컴포넌트 기반 시스템 내의 단위 컴포넌트 간의 입출력 연관관계 정보를 입력받는 입력부

를 더 포함하는, 소프트웨어 컴포넌트 기반 시스템을 위한 테스트 장치.

**청구항 11**

제9항에 있어서,

상기 프로세서는 상기 입출력 연관관계 그래프를 이용하여 상기 외부 입력 파라미터 연관관계 집합을 생성하는,

소프트웨어 컴포넌트 기반 시스템을 위한 테스트 장치.

**청구항 12**

삭제

**청구항 13**

제9항에 있어서,

상기 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 구조가 제1 유형으로 결정된 경우, 상기 프로세서는 상기 테스트 데이터의 집합 모두를 조합하여 상기 테스트 데이터 조합을 생성하는,

소프트웨어 컴포넌트 기반 시스템을 위한 테스트 장치.

**청구항 14**

제9항에 있어서,

상기 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 구조가 제2 유형으로 결정된 경우, 상기 프로세서는 인디펜던트 유니온(Independent Union) 알고리즘에 따라 상기 외부 입력 파라미터 집합 및 상기 외부 입력 파라미터 연관관계 집합에 기초하여 상기 테스트 데이터 조합을 생성하는,

소프트웨어 컴포넌트 기반 시스템을 위한 테스트 장치.

**청구항 15**

제9항에 있어서,

상기 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 구조가 제3 유형으로 결정된 경우, 상기 프로세서는 하이브리드오더(HybridOrder) 알고리즘에 따라 상기 외부 입력 파라미터 집합 및 상기 외부 입력 파라미터 연관관계 집합에 기초하여 상기 테스트 데이터 조합을 생성하는,

소프트웨어 컴포넌트 기반 시스템을 위한 테스트 장치.

**청구항 16**

소프트웨어 컴포넌트 기반 시스템을 위한 테스트 방법에 있어서,

소프트웨어 컴포넌트 기반 시스템 내의 단위 컴포넌트 간의 입출력 연관관계 정보에 기초하여 상기 입출력 연관관계 그래프를 생성하는 단계;

상기 소프트웨어 컴포넌트 기반 시스템의 테스트 데이터의 집합을 포함하는 외부 입력 파라미터 집합을 생성하는 단계;

상기 입출력 연관관계 그래프를 이용하여 외부 입력 파라미터 연관관계 집합을 생성하는 단계;

상기 외부 입력 파라미터 연관관계 집합에 기초하여 상기 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 구조의 유형을 결정하는 단계;

상기 결정된 입출력 연관관계 구조의 유형에 따라 상이한 알고리즘에 의해 테스트 데이터 조합을 생성하는 단계를 포함하고,

상기 입출력 연관관계 구조의 유형은, 일부의 외부 출력에 대하여 모든 외부 입력이 영향을 주는 연관관계 구조를 나타내는 제1 유형, 각각의 외부 출력에 대하여 서로 독립적인 외부 입력이 영향을 주는 연관관계 구조를 나타내는 제2 유형 및 각각의 외부 출력에 대하여 서로 교집합을 가지는 외부 입력이 영향을 주는 연관관계 구조를 나타내는 제3 유형을 포함하는, 소프트웨어 컴포넌트 기반 시스템을 위한 테스트 방법.

**청구항 17**

삭제

**청구항 18**

제16항에 있어서,

상기 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 구조가 제1 유형으로 결정된 경우, 상기 테스트 데이터 조합을 생성하는 단계는, 상기 테스트 데이터의 집합 모두를 조합하여 상기 테스트 데이터 조합을 생성하는 단계를 포함하는,

소프트웨어 컴포넌트 기반 시스템을 위한 테스트 방법.

**청구항 19**

제16항에 있어서,

상기 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 구조가 제2 유형으로 결정된 경우, 상기 테스트 데이터 조합을 생성하는 단계는, 인디펜던트 유니온(Independent Union) 알고리즘에 따라 상기 외부 입력 파라미터 집합 및 상기 외부 입력 파라미터 연관관계 집합에 기초하여 상기 테스트 데이터 조합을 생성하는 단계를 포함하는,

소프트웨어 컴포넌트 기반 시스템을 위한 테스트 방법.

**청구항 20**

제16항에 있어서,

상기 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 구조가 제3 유형으로 결정된 경우, 상기 테스트 데이터 조합을 생성하는 단계는, 하이브리드오더(HybridOrder) 알고리즘에 따라 상기 외부 입력 파라미터 집합 및 상기 외부 입력 파라미터 연관관계 집합에 기초하여 상기 테스트 데이터 조합을 생성하는 단계를 포함하는,

소프트웨어 컴포넌트 기반 시스템을 위한 테스트 방법.

**발명의 설명**

**기술분야**

[0001] 본 발명은 소프트웨어 컴포넌트 기반 시스템의 소프트웨어 테스트를 위한 장치 및 방법에 관한 것이다. 보다 구체적으로, 본 발명은 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계를 기반으로 소프트웨어 컴포넌트 기반 시스템의 테스트를 위한 테스트 데이터의 조합을 생성하는 장치 및 방법에 관한 것이다.

**배경기술**

[0002] 로봇 기술이 다양해지고 시스템의 규모와 복잡도가 증가함에 따라 컴포넌트 기반 개발(Component Based Development)의 유용성이 대두되고 있다. 컴포넌트 기반 개발은 특정 프레임워크 상에서 실행되는 부품화된 컴포넌트를 바탕으로 이를 조립하여 더 큰 컴포넌트를 만들거나 애플리케이션을 개발하는 기법이다.

[0003] 컴포넌트 기반 개발을 통해 개발된 소프트웨어 컴포넌트 기반 시스템(Software Component Based System)은 재사용 가능한 여러 컴포넌트들로 구성되며, 각 컴포넌트들은 입출력 인터페이스를 통해서 연결되고 일부 입출력은 시스템의 외부로 노출된다. 소프트웨어 컴포넌트 기반 시스템은 각 컴포넌트 내부의 입출력 연관관계(Intra I/O relationship) 및 컴포넌트 간의 입출력 연관관계(Inter I/O relationship)를 가지며, 컴포넌트 내부의 입출력 연관관계는 각 컴포넌트를 개발하는 개발자에 의해 정의될 수 있고, 컴포넌트 간의 입출력 연관관계는 각 컴포넌트들을 조립하여 시스템을 개발하는 개발자에 의해 정의될 수 있다.

[0004] 소프트웨어의 품질 관리에 가장 널리 이용되는 방법은 소프트웨어 테스트로서, 일반적으로 소프트웨어 개발에 소요되는 총 비용의 50% 이상과 총 기간의 50% 정도가 소프트웨어의 테스트 작업에 할당되고 있다. 여러 가지 소프트웨어 테스트 기법 중에서, 명세 기반 테스트 기법은 소스 코드, 즉 내부 구조를 고려하지 않고 명세 정보를 바탕으로 테스트를 수행하는 기법이다. 기본적인 테스트 수행 절차는 일반적으로 테스트 명세 정보 작성 단계, 테스트 데이터 생성 단계, 테스트 케이스(테스트 데이터의 조합) 생성 단계, 예상 결과 값 생성 단계, 그리고 테스트 수행 단계로 이루어진다. 이때 소프트웨어 컴포넌트 기반 시스템의 정확한 동작을 검증하기 위한 테스트 데이터와 테스트 케이스를 수동으로 생성하는 경우 많은 시간과 비용이 소비된다.

[0005] 테스트 데이터 생성을 위한 종래의 명세 기반 테스트 기법들은 테스트 대상 소프트웨어 컴포넌트 기반 시스템의 각 입력 파라미터가 모두 독립적이라는 가정 하에 명세 정보를 기반으로 각 입력마다 동등 클래스, 경계 값, 임의의 값 등을 이용하여 테스트 데이터를 생성한다. 이 기법들은 다양한 분야에 활용 될 수 있지만 각 입력들 간의 연관관계를 고려하여 테스트하기에는 적합하지 않다.

[0006] 각 입력 파라미터의 테스트 데이터들을 서로 조합하여 테스트 케이스를 생성할 때 모든 조합의 수를 테스트 하는 경우(exhaustive testing), 테스트 케이스의 폭발적인 증가로 인하여 테스트에 많은 시간과 비용이 소비된다. 조합 테스트 기법(Combinatorial testing)은 각 입력 파라미터 간의 교호작용(interaction)으로

인해 발생하는 결함(fault)을 찾을 수 있는 테스트 기법으로서, 불필요한 테스트 데이터 조합을 제거하여 테스트 케이스의 수를 줄이는 기법이다. 하지만, 종래의 조합 테스트 기법들은 테스트 대상 시스템의 입력 파라미터들 간의 교호작용을 미리 가정하여 축소된 테스트 케이스 집합을 생성하며, 따라서 실제 소프트웨어 컴포넌트 기반 시스템의 입력 파라미터들 간의 교호작용이 미리 가정된 것과 다르면 불필요한 테스트 케이스가 존재하거나 결함을 발견할 수 있는 테스트 케이스가 존재하지 않을 수도 있다. 이러한 문제를 해결하고자 실제 입출력 연관관계를 고려하여 각 테스트 데이터들을 조합하는 입출력 연관관계 기반 조합 테스트 기법들이 제안되었다. 그러나, 종래의 입출력 연관관계 기반 조합 테스트 기법들은 실제 입출력 연관관계를 사전에 가정하고 있으며, 불필요한 테스트 케이스를 생성하거나 시간 복잡도가 우수하지 못하다는 단점을 가지고 있다. 그러므로, 소프트웨어 컴포넌트 기반 시스템의 실제 입출력 연관관계를 고려하여 최적의 테스트 케이스를 생성하기 위한 방법이 필요하다.

**발명의 내용**

**과제의 해결 수단**

- [0007] 본 발명은 소프트웨어 컴포넌트 기반 시스템을 위한 입출력 연관관계 기반 적응형 조합 테스트 기법을 제안한다. 본 발명에서는 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 구조에 있어서, 세 가지 유형의 시스템 입출력 연관관계 구조를 정의한다. 본 발명에서는 각 컴포넌트 내부의 입출력 연관관계 및 컴포넌트 간의 입출력 연관관계에 기초하여 소프트웨어 컴포넌트 기반 시스템의 입력 파라미터 연관관계 집합을 생성하고, 시스템의 입출력 연관관계 구조에 따라 각 입력 파라미터의 테스트 데이터들을 적응적(adaptive)으로 조합하여 테스트 케이스를 생성한다. 단위 컴포넌트 간의 입출력 연관관계는 미리 주어지고, 테스트의 대상이 되는 소프트웨어 컴포넌트 기반 시스템은 결정론적(deterministic)이라고 가정한다.
- [0008] 본 발명의 일실시예에 따른 소프트웨어 컴포넌트 기반 시스템을 위한 테스트 장치는, 소프트웨어 컴포넌트 기반 시스템의 테스트 데이터의 집합을 포함하는 외부 입력 파라미터 집합 및 외부 입력 파라미터 연관관계 집합을 생성하고, 상기 외부 입력 파라미터 연관관계 집합에 기초하여 상기 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 구조의 유형을 결정하고, 상기 결정된 입출력 연관관계 구조의 유형에 따라 상이한 알고리즘에 의해 테스트 데이터 조합을 생성하는 프로세서를 포함할 수 있다.
- [0009] 상기 테스트 장치는, 상기 소프트웨어 컴포넌트 기반 시스템의 외부 입력 파라미터, 상기 소프트웨어 컴포넌트 기반 시스템의 외부 출력 파라미터 및 상기 소프트웨어 컴포넌트 기반 시스템 내의 단위 컴포넌트 간의 입출력 연관관계 정보 및 상기 단위 컴포넌트 내부의 입출력 연관관계 정보를 입력받는 입력부를 더 포함할 수 있다.
- [0010] 상기 프로세서는 상기 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 그래프를 생성하고, 상기 입출력 연관관계 그래프를 이용하여 상기 외부 입력 파라미터 연관관계 집합을 생성할 수 있다.
- [0011] 상기 프로세서는 상기 소프트웨어 컴포넌트 기반 시스템 내의 단위 컴포넌트 간의 입출력 연관관계 정보 및 상기 단위 컴포넌트 내부의 입출력 연관관계 정보에 기초하여 상기 입출력 연관관계 그래프를 생성할 수 있다.
- [0012] 상기 입출력 연관관계 구조의 유형은, 일부의 외부 출력에 대하여 모든 외부 입력이 영향을 주는 연관관계 구조를 나타내는 제1 유형, 각각의 외부 출력에 대하여 서로 독립적인 외부 입력이 영향을 주는 연관관계 구조를 나타내는 제2 유형 및 각각의 외부 출력에 대하여 서로 교집합을 가지는 외부 입력이 영향을 주는 연관관계 구조를 나타내는 제3 유형을 포함할 수 있다.
- [0013] 상기 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 구조가 제1 유형으로 결정된 경우, 상기 프로세서는 상기 테스트 데이터의 집합 모두를 조합하여 상기 테스트 데이터 조합을 생성할 수 있다.
- [0014] 상기 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 구조가 제2 유형으로 결정된 경우, 상기 프로세서는 인디펜던트 유니온(Independent Union) 알고리즘에 따라 상기 외부 입력 파라미터 집합 및 상기 외부 입력 파라미터 연관관계 집합에 기초하여 상기 테스트 데이터 조합을 생성할 수 있다.
- [0015] 상기 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 구조가 제3 유형으로 결정된 경우, 상기 프로세서는 하이브리드오더(HybridOrder) 알고리즘에 따라 상기 외부 입력 파라미터 집합 및 상기 외부 입력 파라미터 연관관계 집합에 기초하여 상기 테스트 데이터 조합을 생성할 수 있다.
- [0016] 본 발명의 또 다른 일실시예에 따른 소프트웨어 컴포넌트 기반 시스템을 위한 테스트 장치는, 소프트웨어 컴포넌트 기반 시스템 내의 단위 컴포넌트 간의 입출력 연관관계 정보에 기초하여 상기 입출력 연관관계 그래프를

생성하고, 상기 소프트웨어 컴포넌트 기반 시스템의 테스트 데이터의 집합을 포함하는 외부 입력 파라미터 집합 및 외부 입력 파라미터 연관관계 집합을 생성하고, 상기 외부 입력 파라미터 연관관계 집합에 기초하여 상기 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 구조의 유형을 결정하고, 상기 결정된 입출력 연관관계 구조의 유형에 따라 상이한 알고리즘에 의해 테스트 데이터 조합을 생성하는 프로세서를 포함할 수 있다.

- [0017] 상기 테스트 장치는, 상기 소프트웨어 컴포넌트 기반 시스템의 외부 입력 파라미터, 상기 소프트웨어 컴포넌트 기반 시스템의 외부 출력 파라미터 및 상기 소프트웨어 컴포넌트 기반 시스템 내의 단위 컴포넌트 간의 입출력 연관관계 정보를 입력받는 입력부를 더 포함할 수 있다.
- [0018] 상기 프로세서는 상기 입출력 연관관계 그래프를 이용하여 상기 외부 입력 파라미터 연관관계 집합을 생성할 수 있다.
- [0019] 상기 입출력 연관관계 구조의 유형은, 일부의 외부 출력에 대하여 모든 외부 입력이 영향을 주는 연관관계 구조를 나타내는 제1 유형, 각각의 외부 출력에 대하여 서로 독립적인 외부 입력이 영향을 주는 연관관계 구조를 나타내는 제2 유형 및 각각의 외부 출력에 대하여 서로 교집합을 가지는 외부 입력이 영향을 주는 연관관계 구조를 나타내는 제3 유형을 포함할 수 있다.
- [0020] 상기 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 구조가 제1 유형으로 결정된 경우, 상기 프로세서는 상기 테스트 데이터의 집합 모두를 조합하여 상기 테스트 데이터 조합을 생성할 수 있다.
- [0021] 상기 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 구조가 제2 유형으로 결정된 경우, 상기 프로세서는 인디펜던트 유니온(Independent Union) 알고리즘에 따라 상기 외부 입력 파라미터 집합 및 상기 외부 입력 파라미터 연관관계 집합에 기초하여 상기 테스트 데이터 조합을 생성할 수 있다.
- [0022] 상기 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 구조가 제3 유형으로 결정된 경우, 상기 프로세서는 하이브리드오더(HybridOrder) 알고리즘에 따라 상기 외부 입력 파라미터 집합 및 상기 외부 입력 파라미터 연관관계 집합에 기초하여 상기 테스트 데이터 조합을 생성할 수 있다.
- [0023] 본 발명의 또 다른 일실시예에 따른 소프트웨어 컴포넌트 기반 시스템을 위한 테스트 방법은, 소프트웨어 컴포넌트 기반 시스템 내의 단위 컴포넌트 간의 입출력 연관관계 정보에 기초하여 상기 입출력 연관관계 그래프를 생성하는 단계; 상기 소프트웨어 컴포넌트 기반 시스템의 테스트 데이터의 집합을 포함하는 외부 입력 파라미터 집합을 생성하는 단계; 상기 입출력 연관관계 그래프를 이용하여 외부 입력 파라미터 연관관계 집합을 생성하는 단계; 상기 외부 입력 파라미터 연관관계 집합에 기초하여 상기 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 구조의 유형을 결정하는 단계; 상기 결정된 입출력 연관관계 구조의 유형에 따라 상이한 알고리즘에 의해 테스트 데이터 조합을 생성하는 단계를 포함할 수 있다.
- [0024] 상기 입출력 연관관계 구조의 유형은, 일부의 외부 출력에 대하여 모든 외부 입력이 영향을 주는 연관관계 구조를 나타내는 제1 유형, 각각의 외부 출력에 대하여 서로 독립적인 외부 입력이 영향을 주는 연관관계 구조를 나타내는 제2 유형 및 각각의 외부 출력에 대하여 서로 교집합을 가지는 외부 입력이 영향을 주는 연관관계 구조를 나타내는 제3 유형을 포함할 수 있다.
- [0025] 상기 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 구조가 제1 유형으로 결정된 경우, 상기 테스트 데이터 조합을 생성하는 단계는, 상기 테스트 데이터의 집합 모두를 조합하여 상기 테스트 데이터 조합을 생성하는 단계를 포함할 수 있다.
- [0026] 상기 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 구조가 제2 유형으로 결정된 경우, 상기 테스트 데이터 조합을 생성하는 단계는, 인디펜던트 유니온(Independent Union) 알고리즘에 따라 상기 외부 입력 파라미터 집합 및 상기 외부 입력 파라미터 연관관계 집합에 기초하여 상기 테스트 데이터 조합을 생성하는 단계를 포함할 수 있다.
- [0027] 상기 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 구조가 제3 유형으로 결정된 경우, 상기 테스트 데이터 조합을 생성하는 단계는, 하이브리드오더(HybridOrder) 알고리즘에 따라 상기 외부 입력 파라미터 집합 및 상기 외부 입력 파라미터 연관관계 집합에 기초하여 상기 테스트 데이터 조합을 생성하는 단계를 포함할 수 있다.

**도면의 간단한 설명**

- [0028] 도 1은 일실시예에 따른 소프트웨어 컴포넌트 기반 시스템을 나타낸다.

- 도 2a는 일실시예에 따른 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 구조의 일 유형을 나타낸다.
- 도 2b는 일실시예에 따른 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 구조의 일 유형을 나타낸다.
- 도 2c는 일실시예에 따른 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 구조의 일 유형을 나타낸다.
- 도 3은 일실시예에 따른 소프트웨어 컴포넌트 기반 시스템의 테스트를 위한 테스트 케이스 생성 동작을 설명하기 위한 순서도이다.
- 도 4는 일실시예에 따른 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 방향 그래프의 생성 알고리즘을 나타낸다.
- 도 5a는 일실시예에 따른 소프트웨어 컴포넌트 기반 시스템의 단위 컴포넌트 내부의 입출력 연관관계 정보 및 단위 컴포넌트 간의 입출력 연관관계 정보를 이용하여 생성된 입출력 연관관계 방향 그래프를 나타낸다.
- 도 5b는 일실시예에 따른 소프트웨어 컴포넌트 기반 시스템의 단위 컴포넌트 간의 입출력 연관관계 정보를 이용하여 생성된 입출력 연관관계 방향 그래프를 나타낸다.
- 도 6은 일실시예에 따른 소프트웨어 컴포넌트 기반 시스템의 테스트 케이스를 생성하는 인디펜던트 유니온 알고리즘을 나타낸다.
- 도 7는 일실시예에 따른 소프트웨어 컴포넌트 기반 시스템의 테스트 케이스를 생성하는 하이브리드오더 알고리즘을 나타낸다.
- 도 8은 일실시예에 따른 소프트웨어 컴포넌트 기반 시스템을 나타낸다.
- 도 9는 일실시예에 따른 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 방향 그래프를 나타낸다.
- 도 10은 일실시예에 따른 소프트웨어 컴포넌트 기반 시스템을 나타낸다.
- 도 11은 일실시예에 따른 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 방향 그래프를 나타낸다.

**발명을 실시하기 위한 구체적인 내용**

[0029] 이하, 실시예들을 첨부된 도면을 참조하여 상세하게 설명한다. 그러나, 본 발명이 실시예들에 의해 제한되는 것은 아니다.

[0030] 도 1은 일실시예에 따른 소프트웨어 컴포넌트 기반 시스템(Software Component Based System; SCBS)을 나타낸다. 도 1에 도시된 바와 같이, 소프트웨어 컴포넌트 기반 시스템은 단위 컴포넌트들로 구성되며, 각 단위 컴포넌트들은 입출력 파라미터를 통해 서로 연결되어 동작하거나 독립적으로 동작한다. 단위 컴포넌트의 일부 입출력 파라미터는 소프트웨어 컴포넌트 기반 시스템의 외부로 노출되는데, 이와 같은 입출력 파라미터를 소프트웨어 컴포넌트 기반 시스템의 외부 입출력 파라미터라고 한다. 입출력 연관관계의 관점에서 소프트웨어 컴포넌트 기반 시스템은 각 단위 컴포넌트의 내부 입출력 연관관계(Intra I/O relationship)와 단위 컴포넌트 간의 입출력 연관관계(Inter I/O relationship)를 가진다.

[0031] 일실시예에 따른 소프트웨어 컴포넌트 기반 시스템(100)은 상호 연결된 복수의 단위 컴포넌트들( $C_1, C_2, C_3, C_4$ )을 포함할 수 있다. 구체적으로, CC는 소프트웨어 컴포넌트 기반 시스템을 구성하는 모든 단위 컴포넌트의 집합을 의미하고,  $C_i$ 는 소프트웨어 컴포넌트 기반 시스템을 구성하는  $i$ 번째 단위 컴포넌트를 의미한다. 또한,  $x_{i,j}$  및  $y_{i,k}$ 는 각각  $C_i$ 의  $j$ 번째 입력 파라미터 및  $k$ 번째 출력 파라미터를 의미한다.  $C_{in}^i$  및  $C_{out}^i$ 은 각각  $C_i$ 의 모든 입력 파라미터의 집합 및 모든 출력 파라미터의 집합을 의미하고,  $EXT_{in}(C_i)$  및  $EXT_{out}(C_i)$ 은 각각  $C_i$ 의 소프트웨어 컴포넌트 기반 시스템의 외부 입력 파라미터의 집합 및 외부 출력 파라미터의 집합을 의미한다. 예를 들어, 도 1에 도시된 소프트웨어 컴포넌트 기반 시스템(100)에서  $C_{in}^1 = \{ x_{1,1}, x_{1,2}, x_{1,3} \}$ 이고  $EXT_{in}(C_1) = \{ x_{1,1}, x_{1,2} \}$ 이다.  $CC_{in}$  및  $CC_{out}$ 은 각각 소프트웨어 컴포넌트 기반 시스템의 모든 외부 입력 파라미터의 집합 및 모든 외부 출력 파라미터의 집합을 의미한다.  $RP_{intra}$ 는 모든 각  $C^i$  내부의 입출력 연관관계 집합인  $C_{rp}^i$ 의 합집합을 의미하고,  $RP_{inter}$ 는 컴포넌트 간의 입출력 연관관계의 집합을 의미하며, RP는  $RP_{intra}$ 와  $RP_{inter}$ 의 합

집합을 의미한다.

[0032] 이상의 정의에 따라, 소프트웨어 컴포넌트 기반 시스템은 수학적 식 1과 같이 정의될 수 있다.

[0033] [수학적 식 1]

[0034] 
$$SCBS = (CC, RP)$$

[0035] 여기서,  $CC = \bigcup_{i=1}^k C_i$ ,  $RP = RP_{intra} \cup RP_{inter}$  이고,  $C_{in}^i = \{x_{i,1}, x_{i,2}, \dots, x_{i,n}\}$ ,  $C_{out}^i = \{y_{i,1}, y_{i,2}, \dots, y_{i,m}\}$ ,  
 $C_{rp}^i = C_{in}^i \times C_{out}^i$ ,  $CC_{in} = \bigcup_{C_i \in CC} EXT_{in}(C_i)$ ,  $CC_{out} = \bigcup_{C_i \in CC} EXT_{out}(C_i)$ ,  $RP_{intra} = \bigcup_{i=1}^k C_{rp}^i$ ,  $RP_{inter} = \bigcup_{i,j} (C_{out}^i \times C_{in}^j)$  이다.

[0036] 소프트웨어 컴포넌트 기반 시스템을 테스트하기 위해서는  $CC_{in}$ 에 속한 각 입력 파라미터에 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  $T_{in}$ 은 수학적 식 2와 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 p는  $CC_{in}$ 의 크기 ( $|CC_{in}|$ )이다. 그리고  $T_{in}$ 의 크기( $|T_{in}|$ )는 수학적 식 3과 같이 각 입력 파라미터들의 테스트 데이터 집합 크기 ( $|D(x_i)|$ )의 곱으로 계산된다.

[0037] [수학적 식 2]

[0038] 
$$T_{in} = D(x_1) \times D(x_2) \times \dots \times D(x_p).$$

[0039] [수학적 식 3]

[0040] 
$$|T_{in}| = |D(x_1)| * |D(x_2)| * \dots * |D(x_p)|.$$

[0041] 소프트웨어 컴포넌트 기반 시스템을 테스트 할 때, 각 입력 파라미터의 모든 테스트 데이터를 조합하여 생성한 테스트 케이스 집합( $T_{in}$ )의 크기가 클수록 테스트에 소요되는 시간 및 비용이 증가하게 된다. 본 발명에서는 전체 테스트 케이스 집합( $T_{in}$ )의 부분집합이며, 이로부터 전체 테스트 데이터 집합의 입력에 대한 출력을 나타내는 전체 예상 결과 값을 도출해 낼 수 있도록 생성되는 축소된 테스트 케이스 집합( $T_{in}^{red}$ )을 생성한다.

[0042] 소프트웨어 컴포넌트 기반 시스템의 외부 입출력 파라미터의 측면에서 살펴보면, 외부 입력 파라미터의 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 외부 입력 파라미터에 대한 테스트 데이터 집합이 될 수 있다. 각 테스트 데이터 집합을 포함하는 외부 입력 파라미터의 집합(F)는  $F = \{f_1, f_2, \dots, f_n\}$ 으로 정의되며, 여기서  $n = |CC_{in}|$  이 된다. 집합 F에 속한 각 외부 입력 파라미터  $f_i = (x_i, D(x_i))$ 로 구성되며, 여기서  $x_i \in CC_{in}$  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 파라미터 간의 연관관계를 나타내는 외부 입력 파라미터 연관관계 집합(R)은  $R = \{r_1, r_2, \dots, r_m\}$ 으로 정의되며, 집합 R에 속한 각 연관관계  $r_i = \{f_1, f_2, \dots, f_k\}$ 는 외부 입력 파라미터들로 구성된다.

[0043] 도 2a 내지 도 2c를 참조하면, 예시적인 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 구조의 유형을 나타낸다. 본 발명에서는 입출력 연관관계 구조의 유형에 따라 각 입력 파라미터의 테스트 데이터들을 적응적으로 조합하여 테스트 케이스를 생성할 수 있다.

[0044] 도 2a는 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 구조의 일 유형으로서 모든 연관관계 구조(제1 유

형)를 나타낸다. 도 2a에 도시된 바와 같이, 모든 연관관계 구조는 특정 외부 출력 파라미터에 대해 모든 외부 입력 파라미터가 영향을 주는 경우이다. 각 외부 출력 파라미터에 영향을 주는 외부 입력 파라미터의 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, 여기서  $X(y_2)$ 와  $X(y_3)$ 는  $X(y_1)$ 의 부분 집합이므로 삭제되고, 결과적으로 외부 입력 파라미터 연관관계 집합  $R = \{r_1\}$ ,  $r_1 = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$ 이 된다. 즉, 모든 연관관계 구조에서 집합 R의 크기는 1이 된다.

[0045] 도 2b는 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 구조의 일 유형으로서 독립된 연관관계 구조(제2 유형)를 나타낸다. 도 2b에 도시된 바와 같이, 독립된 연관관계 구조는 각 외부 출력 파라미터에 서로 상이한 외부 입력 파라미터가 영향을 주는 경우이다. 각 외부 출력 파라미터에 영향을 주는 외부 입력 파라미터의 집합을 구하면  $X(y_1) = \{x_1, x_2, x_4\}$ ,  $X(y_2) = \{x_3, x_5\}$ ,  $X(y_3) = \{x_6, x_7\}$ 이 되며, 외부 입력 파라미터 연관관계 집합  $R = \{r_1, r_2, r_3\}$ ,  $r_1 = \{x_1, x_2, x_4\}$ ,  $r_2 = \{x_3, x_5\}$ ,  $r_3 = \{x_6, x_7\}$ 이 된다. 즉, 독립된 연관관계 구조에서 각 연관관계는 서로 독립적이다.

[0046] 도 2c는 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 구조의 일 유형으로서 교집합 연관관계 구조(제3 유형)를 나타낸다. 도 2c에 도시된 바와 같이, 교집합 연관관계 구조는 일부 외부 입력 파라미터가 몇몇 출력 파라미터에 동시에 영향을 주는 경우이다. 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, 외부 입력 파라미터 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 된다. 즉, 교집합 연관관계 구조에서 각 연관관계 사이에는 서로 교집합이 존재한다.

[0047] 이하에서는 도 3을 참조하여, 일 실시예에 따른 소프트웨어 컴포넌트 기반 시스템의 테스트를 위한 테스트 케이스 생성 동작에 대하여 구체적으로 설명한다. 일 실시예에 따른 테스트 케이스 생성 동작은 입력으로서 입출력 연관관계 정보(RP), 외부 입력 파라미터의 집합(CC<sub>in</sub>) 및 외부 출력 파라미터의 집합(CC<sub>out</sub>)을 가지며, 출력으로서 축소된 테스트 케이스의 집합(T<sub>in</sub><sup>red</sup>)을 가진다.

[0048] 단계 310에서, 테스트 장치는 입출력 연관관계 정보(RP), 외부 입력 파라미터의 집합(CC<sub>in</sub>) 및 외부 출력 파라미터의 집합(CC<sub>out</sub>)을 입력받을 수 있다.

[0049] 단계 320에서, 테스트 장치는 입출력 연관관계 정보(RP), 외부 입력 파라미터의 집합(CC<sub>in</sub>) 및 외부 출력 파라미터의 집합(CC<sub>out</sub>)에 기초하여 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 그래프(I/O relationship directed graph; IORG)를 생성할 수 있다. V가 그래프 상의 정점으로서 모든 입력 파라미터 및 모든 출력 파라미터를 의미하고 E가 그래프 상의 에지(edge)로서 입출력 연관관계를 의미할 때, 입출력 연관관계 방향 그래프는 IORG = (V, E)로 구성된다.

[0050] 입출력 연관관계 그래프는 예를 들어 도 4의 알고리즘에 의하여 생성될 수 있다. 종래의 알고리즘에 따르면 소프트웨어 컴포넌트 기반 시스템의 각 컴포넌트 내부의 입출력 연관관계(RP<sub>intra</sub>)와 컴포넌트 간의 입출력 연관관계(RP<sub>inter</sub>)가 모두 주어졌을 때 IORG를 생성할 수 있었지만, 도 4의 알고리즘에 따르면 컴포넌트 간의 입출력 연관관계(RP<sub>inter</sub>)만 주어진 경우에도 IORG를 생성할 수 있다.

[0051] 도 4의 알고리즘은 먼저, IORG의 V 및 E를 초기화한다(line 1, 2). 이어서, CC<sub>in</sub>의 각 원소, 즉 외부 입력 파라미터 x에 대하여, IORG의 V에 x를 추가하고(line 4), x에 연관된 V 및 E를 생성하기 위해 x를 인자로 하여 함수 GenerateIOReGraph()를 호출하고(line 5), 이 함수의 결과를 IORG에 추가한다(line 6). 함수 GenerateIOReGraph()의 출력 결과 G는 CC<sub>in</sub>의 각 외부 입력 파라미터 x와 연관 있는 소프트웨어 컴포넌트 기반 시스템의 모든 입출력 파라미터들 간의 연관관계를 나타낸다.

[0052] 함수 GenerateIOReGraph()는 재귀적 함수로서, 함수의 상세한 동작은 다음과 같다. RP에 속한 컴포넌트 내부 입출력 연관관계 집합(RP<sub>intra</sub>)이 공집합이 아닐 경우, RP<sub>intra</sub>에서 입력 inp와 관련이 있는 내부 출력 파라미터(intra-output) 집합을 구해 RO에 저장한다(line 11, 12). 한편, RP<sub>intra</sub>가 공집합일 경우, 즉 단위 컴포넌트들

의 내부 입출력 연관관계가 알려지지 않은 경우, 입력 inp를 포함하는 해당 단위 컴포넌트들의 모든 내부 출력 집합을 구해 R0에 저장한다(line 13, 14, 15). 그 후, R0에 속한 각 내부 출력 파라미터 rout에 대하여, rout을 그래프의 정점 집합(G.V)에 저장하고(line 17), 입력 inp에서 rout으로의 에지를 추가하여 에지 집합(G.E)에 저장한다(line 18). 이어서, RP에 속한 컴포넌트 간 입출력 연관관계 집합(RP<sub>inter</sub>)에서 rout과 관련 있는 입력 파라미터(inter-inputs) 집합을 구해 RI에 저장한다(line 19). RI가 공집합이거나 rout이 CC<sub>out</sub>의 원소, 즉 외부 출력 파라미터일 경우, 더 이상 연관된 입출력 파라미터를 검색하지 않는다(line 20). 그렇지 않은 경우, RI에 속한 각 입력 파라미터 rin에 대하여, rin을 그래프의 정점 집합(G.V)에 저장하고(line 22), 출력 rout에서 rin으로의 에지를 추가하여 에지 집합(G.E)에 저장한다(line 23). 그 후, rin을 인자로 하여 함수 GenerateIOReGraph()를 재귀적으로 호출한다(line 24).

[0053] 도 5a 및 도 5b는 도 1의 소프트웨어 컴포넌트 기반 시스템에 대하여 도 4의 알고리즘을 통해 생성한 IORG를 나타낸다. 도시된 바와 같이, 도 1의 소프트웨어 컴포넌트 기반 시스템의 IORG는 10개의 입력 노드와 8개의 출력 노드 및 각 노드들 간의 에지들로 구성된다.

[0054] 도 5a는 소프트웨어 컴포넌트 기반 시스템의 각 단위 컴포넌트 간의 입출력 연관관계 집합(RP<sub>inter</sub>)과 각 단위 컴포넌트 내부의 입출력 연관관계 집합(RP<sub>intra</sub>)이 모두 주어졌을 때 생성된 실제 IORG이며, 도 5b는 소프트웨어 컴포넌트 기반 시스템의 각 단위 컴포넌트 간의 입출력 연관관계 집합(RP<sub>inter</sub>)만이 주어졌을 때 생성된 가상의 IORG이다. 도 5b에 도시된 바와 같이, RP<sub>intra</sub>가 주어지지 않은 경우 도 4의 알고리즘의 line 13, 14, 15에 의해 해당 단위 컴포넌트의 각 입력 파라미터는 모든 내부 출력 파라미터들과 가상의 연관관계를 형성하여 IORG를 생성한다. 본 발명에서는 이러한 방식으로 생성한 IORG를 통해 소프트웨어 컴포넌트 기반 시스템의 각 외부 출력 파라미터에 영향을 주는 외부 입력 파라미터들을 식별할 수 있다.

[0055] 단계 330에서, 테스트 장치는 외부 입력 파라미터의 집합(F) 및 외부 입력 파라미터 연관관계 집합(R)을 생성할 수 있다.

[0056] 테스트 장치는 소프트웨어 컴포넌트 기반 시스템의 각 외부 입력 파라미터의 테스트 데이터를 명세 기반 테스트 기법을 이용하여 생성한 후 집합 F를 생성할 수 있다. 예를 들어, 도 1의 소프트웨어 컴포넌트 기반 시스템의 집합 F는  $F = \{f_1, f_2, f_3, f_4, f_5, f_6, f_7\}$ 이며,  $f_1 = (x_{1,1}, D(x_{1,1}))$ ,  $f_2 = (x_{1,2}, D(x_{1,2}))$ ,  $f_3 = (x_{2,1}, D(x_{2,1}))$ ,  $f_4 = (x_{2,2}, D(x_{2,2}))$ ,  $f_5 = (x_{4,1}, D(x_{4,1}))$ ,  $f_6 = (x_{4,2}, D(x_{4,2}))$ ,  $f_7 = (x_{4,3}, D(x_{4,3}))$ 이 된다. 표 1은 도 1의 소프트웨어 컴포넌트 기반 시스템의 각 외부 파라미터에 대한 테스트 데이터를 나타낸다.

표 1

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

[0057]

[0058] 이어서, 집합 R을 생성하기 위해, 테스트 장치는 각 외부 출력 파라미터에 영향을 주는 부분 집합을 포함하지 않는 외부 입력 파라미터들의 집합을 구할 수 있다.

[0059] 예를 들어, 도 5a의 실제 IORG를 이용하여 각 집합을 구하면,  $X(y_{1,1})=\{x_{1,1}, x_{1,2}\}$ ,  $X(y_{1,2})=\{x_{1,2}, x_{2,1}\}$ ,  $X(y_{3,1})=\{x_{2,1}, x_{2,2}\}$ ,  $X(y_{3,2})=\{x_{2,2}\}$ ,  $X(y_{4,1})=\{x_{4,1}, x_{4,2}, x_{4,3}\}$ 이다. 이 때,  $X(y_{3,2}) \subset X(y_{3,1})$ 이 되므로  $X(y_{3,2})$ 의 연관관계는 삭제되어, 결과적인 집합 R은  $R=\{r_1, r_2, r_3, r_4\}$ ,  $r_1=\{f_1, f_2\}$ ,  $r_2=\{f_2, f_3\}$ ,  $r_3=\{f_3, f_4\}$ ,  $r_4=\{f_5, f_6,$

$f_7$ 이 된다. 즉, 각 연관관계 집합  $r_1, r_2, r_3$  간에는 서로 교집합이 존재하고  $r_4$ 만 독립적인 관계가 된다.

[0060]

한편, 도 5b의 가상 IORG를 이용하여 각 집합을 구하면,  $X(y_{1,1})=\{x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2}\}$ ,  $X(y_{1,2})=\{x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2}\}$ ,  $X(y_{3,1})=\{x_{2,1}, x_{2,2}\}$ ,  $X(y_{3,2})=\{x_{2,1}, x_{2,2}\}$ ,  $X(y_{4,1})=\{x_{4,1}, x_{4,2}, x_{4,3}\}$ 이다. 이 때, 각 집합간의 관계를 보면  $X(y_{1,1})=X(y_{1,2})$ ,  $X(y_{3,1})=X(y_{3,2})$ ,  $(X(y_{1,1}), X(y_{1,2})) \supset (X(y_{3,1}), X(y_{3,2}))$ 이므로 부분 집합인  $X(y_{3,1}), X(y_{3,2})$ 은 삭제되어, 결과적인 집합  $R=\{r_1, r_2\}$ 이며,  $r_1=\{f_1, f_2, f_3, f_4\}$ ,  $r_2=\{f_5, f_6, f_7\}$ 이 된다. 즉, 각 연관관계 집합 간에 서로 교집합이 존재하지 않는 서로 독립적인 관계가 된다.

[0061]

단계 340에서, 테스트 장치는 연관관계 집합 R의 크기가 1인지 여부를 판단할 수 있다. 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 구조가 모든 연관관계 구조인 경우, 즉 특정 외부 출력 파라미터에 대해 모든 외부 입력 파라미터가 영향을 주는 경우, 연관관계 집합 R의 크기가 1이 된다. 연관관계 집합 R의 크기가 1인 경우, 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 구조가 제1 유형(모든 연관관계 구조)에 속하는 것으로 결정하고, 단계 350으로 진행할 수 있다. 연관관계 집합 R의 크기가 1이 아닌 경우, 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 구조의 유형을 결정하기 위하여 단계 360으로 진행될 수 있다.

[0062]

단계 350에서, 테스트 장치는 축소된 테스트 케이스 집합( $T_{in}^{red}$ )을 생성하지 않고 각 외부 입력 파라미터들의 테스트 데이터를 모두 조합(All combinations)하여 전체 테스트 케이스 집합( $T_{in}$ )을 생성할 수 있다. 이 때,  $T_{in}$

을 생성하기 위한 시간 복잡도는  $O(\prod_{f_i \in F} D(f_i))$  이 된다. 예를 들어, 표 2의 테스트 데이터 집합을 이용하여  $T_{in}$ 을 생성하면 다음과 같이 128개의 테스트 케이스가 생성되며, 표 2는 이러한 방식으로 생성된 전체 테스트 케이스 집합을 나타낸다.

[0063]

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

[0064]

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

표 2

ID	$x_{1,1}$	$x_{1,2}$	$x_{2,1}$	$x_{2,2}$	$x_{4,1}$	$x_{4,2}$	$x_{4,3}$
$t_1$	0	0	0	0	0	0	0
$t_2$	0	0	0	0	0	0	1
$t_3$	0	0	0	0	0	1	0
.....							
$t_{60}$	0	1	1	1	0	1	1
$t_{61}$	0	1	1	1	1	0	0
$t_{62}$	0	1	1	1	1	0	1
.....							
$t_{126}$	1	1	1	1	1	0	1
$t_{127}$	1	1	1	1	1	1	0
$t_{128}$	1	1	1	1	1	1	1

[0065]

[0066]

단계 360에서, 테스트 장치는 연관관계 집합 R에 속한 연관관계가 서로 독립적인지 여부를 판단할 수 있다. 연관관계 집합 R에 속한 연관관계가 서로 독립적인 경우, 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 구조가 제2 유형(독립적 연관관계 구조)에 속하는 것으로 결정하고, 단계 370으로 진행할 수 있다. 연관관계 집합 R에 속한 연관관계가 서로 독립적이지 않은 경우, 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 구조가 제3 유형(교집합 연관관계 구조)에 속하는 것으로 결정하고, 단계 380으로 진행할 수 있다.

[0067]

단계 370에서, 테스트 장치는 인디펜던트 유니온(Independent Union) 알고리즘에 따라 축소된 테스트 케이스 집합( $T_{in}^{red}$ )을 생성할 수 있다. Independent Union 알고리즘은 서로 독립적인 연관관계 집합들을 서로 합하여 축

소된 테스트 케이스 집합( $T_{in}^{red}$ )을 생성하는 알고리즘으로서, 기존의 Union 알고리즘과 달리 "don't care data"를 임의로 생성하여 합하지 않으므로 불필요한 테스트 케이스의 수를 줄일 수 있다.

[0068]

도 6은 Independent Union 알고리즘의 상세 동작을 나타낸다. Independent Union 알고리즘은 입력으로 테스트 데이터 집합을 포함하는 외부 입력 파라미터들의 집합 F 와 독립적인 연관관계 집합 R를 가지며, 출력으로 테스트 케이스 집합  $T_{idt}$ 를 가진다. 본 알고리즘은 먼저  $T_{idt}$ 를 초기화 하고 (line 1), R의 각 연관관계 r마다 여기에 속한 외부 입력 파라미터들만의 테스트 데이터들을 서로 조합하여 테스트 케이스 집합을 생성한다(line 2-10). Independent Union 알고리즘을 보다 구체적으로 설명하면 다음과 같다. 연관관계 r에 속한 외부 입력 파라미터들의 테스트 데이터 조합 수를 계산하여 size에 저장한다(line 3). 만약 size가 현재  $T_{idt}$ 의 크기( $|T_{idt}|$ )보다 크다면, (size- $|T_{idt}|$ ) 개수만큼 "don't care data"를 포함하는 빈 테스트 케이스들을 생성하여  $T_{idt}$ 에 추가 한다(line 4-7). 이어서, r에 속한 외부 입력 파라미터들의 테스트 데이터들을 모두 조합하여  $T_{idt}$ 에 채운다 (line 8). 각 연관관계에 대해 위와 같은 작업을 반복한 후 최종적으로  $T_{idt}$ 를 출력한다(line 10).

$$O\left(\sum_{k=1}^t \left(\prod_{f_i \in r_k} D(f_i)\right)\right)$$

Independent Union 알고리즘의 시간 복잡도는  $O\left(\sum_{k=1}^t \left(\prod_{f_i \in r_k} D(f_i)\right)\right)$  이며, 여기서 t는 R의 크기( $|R|$ )이다.

[0069]

도 5b의 가상 IORG를 이용하여 생성한 집합 R의 경우 독립적 연관관계 구조를 나타내고, 따라서 Independent Union 알고리즘을 통해 테스트 케이스 집합을 구할 수 있다. 예를 들어, 표 3 및 표 4는 Independent Union 알고리즘의 진행에 따른  $T_{idt}$ 를 나타낸다. 여기서, 기호 "-"는 "don't care data"를 의미한다. 먼저, Independent Union 알고리즘에서 초기  $|T_{idt}|=0$ 이 되며, R에 속한  $r=r_1$ 인 경우,  $size=|D(f_1)|*|D(f_2)|*|D(f_3)|*|D(f_4)|=16$  이 되어 16개의 빈 테스트 케이스들을 추가한 후  $r_1$ 에 속한 입력 파라미터들의 테스트 데이터를 서로 조합하여  $T_{idt}$ 에 채운다. 표 4은  $r=r_1$ 일 때 현재  $T_{idt}$ 를 나타낸다. 이어서,  $r=r_2$ 인 경우,  $size=|D(f_5)|*|D(f_6)|*|D(f_7)|=8$ 이 되어  $|T_{idt}|$ 의 크기가 size의 크기보다 크므로  $T_{idt}$ 에 빈 테스트 케이스들을 추가하지 않고,  $r_2$ 에 속한 입력 파라미터들의 테스트 데이터를 서로 조합하여  $T_{idt}$ 에 채운다. 표 5는  $r=r_2$ 일 때의  $T_{idt}$ 를 나타내며, 이로써 16개의 테스트 케이스로 구성된 축소된 테스트 케이스 집합( $T_{in}^{red}$ )이 생성된다.

표 3

$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$
0	0	0	0	-	-	-
0	0	0	1	-	-	-
0	0	1	0	-	-	-
0	0	1	1	-	-	-
0	1	0	0	-	-	-
0	1	0	1	-	-	-
0	1	1	0	-	-	-
0	1	1	1	-	-	-
1	0	0	0	-	-	-
1	0	0	1	-	-	-
1	0	1	0	-	-	-
1	0	1	1	-	-	-
1	1	0	0	-	-	-
1	1	0	1	-	-	-
1	1	1	0	-	-	-
1	1	1	1	-	-	-

[0070]

표 4

ID	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$
$rt_1$	0	0	0	0	0	0	0
$rt_2$	0	0	0	1	0	0	1
$rt_3$	0	0	1	0	0	1	0
$rt_4$	0	0	1	1	0	1	1
$rt_5$	0	1	0	0	1	0	0
$rt_6$	0	1	0	1	1	0	1
$rt_7$	0	1	1	0	1	1	0
$rt_8$	0	1	1	1	1	1	1
$rt_9$	1	0	0	0	-	-	-
$rt_{10}$	1	0	0	1	-	-	-
$rt_{11}$	1	0	1	0	-	-	-
$rt_{12}$	1	0	1	1	-	-	-
$rt_{13}$	1	1	0	0	-	-	-
$rt_{14}$	1	1	0	1	-	-	-
$rt_{15}$	1	1	1	0	-	-	-
$rt_{16}$	1	1	1	1	-	-	-

[0071]

[0072]

단계 380에서, 테스트 장치는 하이브리드오더(HybridOrder) 알고리즘에 따라 축소된 테스트 케이스 집합( $T_{in}^{red}$ )을 생성할 수 있다. HybridOrder 알고리즘은 기존의 ParaOrder 알고리즘을 개선한 것으로서 집합 R에 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 알고리즘이다.

[0073]

도 7은 HybridOrder 알고리즘의 상세 동작을 나타낸다. HybridOrder 알고리즘은 입력으로 테스트 데이터 집합을 포함하는 외부 입력 파라미터들의 집합 F와 연관관계 집합 R를 가지며, 출력으로 축소된 테스트 케이스 집합( $T_{in}^{red}$ )을 가진다. 본 알고리즘은 크게 독립된 연관관계 별로 테스트 데이터를 조합( $T_{idt}$ ) 하는 단계(line 1-14), 독립된 연관관계에 포함되어 있지 않은 입력 파라미터 별로 테스트 데이터를 조합( $T_{in}^{red}$ )하는 단계(line 15-26), 마지막으로  $T_{in}^{red}$ 에  $T_{idt}$ 를 합하는 단계(line 27-32)로 이루어져 있다.

[0074]

HybridOrder 알고리즘을 보다 구체적으로 설명하면 다음과 같다. 먼저 R에 속한 연관관계 중에서 독립된 연관관계 집합을 얻어  $R_{idt}$ 에 저장한다(line 2). 만약  $R_{idt}$ 가 공집합( $\Phi$ )이 아니라면,  $R_{idt}$ 에 속한 외부 입력 파라미터들의 집합( $F_{idt}$ )을 구해 Independent Union 알고리즘을 통해 독립된 테스트 케이스 집합을 생성하여  $T_{idt}$ 에 저장한다(line 4, 5). 이어서, 집합 R에서  $R_{idt}$ 에 속한 연관관계 집합을 삭제하고, 집합 F에서  $F_{idt}$ 에 속한 입력 파라미터들을 삭제한다(line 6, 7). 독립된 연관관계 집합에 대한 테스트 케이스가 생성된 후 집합 R에서 크기가 가장 작은 연관관계 r을 선택하여(SelectRelation()) r에 속한 파라미터들의 테스트 데이터를 서로 조합하여  $T_{in}^{red}$ 에 추가한다(line 9-14). 이 때, 집합 F에서는 r에 속한 파라미터들을 삭제하고, 이미 생성한 입력 파라미터들의 집합  $F_{deal}$ 에는 r에 속한 파라미터들을 추가한다. 이어서, 집합 F에 속한 각 입력 파라미터에서 크기가 가장 작은  $f_i$ 를 선택하여(SelectFactor()),  $f_i$ 의 테스트 데이터를 ParaOrder 알고리즘과 유사한 방식으로  $T_{in}^{red}$ 에 추가한다(line 15-26). 이 때, 기존의 ParaOrder 알고리즘과는 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 RedInters를 생성하고(line 19), RedInters의 각 엘리먼트에 속한 입력파라미터들의 테스트 데이터를 서로 조합하여 CombSet을 생성한다(line 20). 그리고, ParaOrder 알고리즘과 유사한 방식으로 CombSet,  $D(f_i)$ 를

이용하여  $T_{in}^{red}$ 을 가로/세로로 확장한다(line 21-26). 마지막으로,  $T_{idt}$ 의 크기( $|T_{idt}|$ )와  $T_{in}^{red}$ 의 크기( $|T_{in}^{red}|$ )를 비교하여,  $T_{idt}$ 의 크기가 더 크다면 ( $|T_{idt}| - |T_{in}^{red}|$ ) 만큼 빈 테스트 케이스를  $T_{in}^{red}$ 에 추가함으로써  $T_{idt}$ 를 합한다 (line 27-33).

[0075]

HybridOrder 알고리즘의 최악의 시간 복잡도는  $O(\sum_{i=1}^w \prod_{f_i \in r_i} D(f_i) + \sum_{i=1}^z ((D(f_i) \times \sum_{k=1}^m \prod_{f_j \in H_k} D(f_j) \times |T_{ct}|)))$  이 되며, Inters에 부분집합이 존재하지 않아 검색되는 CombSet의 크기가 줄어들며, 독립된 연관관계 집합이 존재할 시에는 더 단순해 질 수 있다.

[0076]

도 5a의 실제 IORG를 이용하여 생성한 집합 R의 경우 교집합 연관관계 구조를 나타내고, 따라서 HybridOrder 알고리즘을 통해 테스트 케이스 집합을 구할 수 있다. 예를 들어, 표 5 내지 9는 HybridOrder 알고리즘의 진행에 따른  $T_{idt}$ 를 나타낸다. 먼저, HybridOrder 알고리즘의 입력 파라미터 R에서 독립적인 연관관계 집합을 구하면  $R_{idt}=\{r_4\}$ 가 되고,  $R_{idt}$ 에 속한 외부 입력 파라미터들의 집합  $F_{idt}=\{f_5, f_6, f_7\}$ 가 된다. 그리고  $R_{idt}$ 와  $F_{idt}$ 를 입력으로 하여 Independent Union 알고리즘을 수행하여 표 5에 나타나는 바와 같이 독립된 테스트 케이스 집합  $T_{idt}$ 를 생성한다. 이어서, F와 R을 재계산하면  $F=\{f_1, f_2, f_3, f_4\}$ ,  $R=\{r_1, r_2, r_3\}$ 이 된다. 재계산된 R에서 임의의 연관관계  $r=r_1$ 을 선택한 경우,  $size=|D(f_1)| * |D(f_2)|=4$ 이므로 표 6에 나타나는 바와 같이  $T_{in}^{red}$ 에 4개의 빈 테스트 케이스들을 생성한 후  $r_1$ 에 속한 입력 파라미터들의 테스트 데이터를 서로 조합하여  $T_{in}^{red}$ 에 채운다. 이어서, F와  $F_{deal}$ 을 재계산하면  $F=\{f_3, f_4\}$ ,  $F_{deal}=\{f_1, f_2\}$ 이 된다. 그리고 집합 F에서  $f_3$ 과  $f_4$ 의 크기는 동일하므로 먼저  $f_1=f_3$ 을 선택하면 교집합의 합  $Inters=\{(f_2, f_3), (f_3)\}$ 가 되고 부분집합을 제거한  $RedInters=\{(f_2, f_3)\}$ 이 된다. 이렇게 생성된  $RedInters$ 를 이용하여 CombSet을 생성하면  $f_2$ 와  $f_3$ 의 테스트 데이터를 서로 조합하므로  $CombSet=\{(0,0), (0,1), (1,0), (1,1)\}$ 이 되고, 이를 이용하여  $T_{in}^{red}$ 을 가로/세로로 확장하면 표 7과 같다. 이어서, F와 R을 재계산하면  $F=\{f_4\}$ ,  $F_{deal}=\{f_1, f_2, f_3\}$ 이 된다. 그리고 집합 F에서  $f_1=f_4$ 을 선택하면 교집합의 합  $Inters=\{(f_3, f_4)\}$ 가 되고 부분 집합이 존재하지 않으므로  $RedInters$ 는  $Inters$ 와 동일하게  $\{(f_3, f_4)\}$ 이 된다. 이렇게 생성된  $RedInters$ 를 이용하여 CombSet을 생성하면  $f_3$ 와  $f_4$ 의 테스트 데이터를 서로 조합하므로  $CombSet=\{(0,0), (0,1), (1,0), (1,1)\}$ 이 되고, 이를 이용하여  $T_{in}^{red}$ 을 가로/세로로 확장하면 표 8과 같다. 마지막으로, 앞서 생성한  $T_{idt}$ 를  $T_{in}^{red}$ 에 추가하면 표 9에 나타나는 바와 같이 최종적으로 8개의 테스트 케이스로 구성된 축소된 테스트 케이스 집합( $T_{in}^{red}$ )이 생성된다.

표 5

$f_1$	$f_2$	$f_3$
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

[0077]

표 6

$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$
0	0	-	-	-	-	-
0	1	-	-	-	-	-
1	0	-	-	-	-	-
1	1	-	-	-	-	-

[0078]

표 7

$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$	RedInters
0	0	0	-	-	-	-	$\{(f_5, f_6)\}$
0	1	0	-	-	-	-	
1	0	1	-	-	-	-	
1	1	1	-	-	-	-	

[0079]

표 8

$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$	RedInters
0	0	0	0	-	-	-	$\{(f_5, f_6)\}$
0	1	0	1	-	-	-	
1	0	1	0	-	-	-	
1	1	1	1	-	-	-	

[0080]

표 9

ID	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$
$rt_1$	0	0	0	0	0	0	0
$rt_2$	0	1	0	1	0	0	1
$rt_3$	1	0	1	0	0	1	0
$rt_4$	1	1	1	1	0	1	1
$rt_5$	-	-	-	-	1	0	0
$rt_6$	-	-	-	-	1	0	1
$rt_7$	-	-	-	-	1	1	0
$rt_8$	-	-	-	-	1	1	1

[0081]

[0082]

이와 같이, 본 발명에서는 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 구조의 유형(모든 연관관계, 독립적 연관관계, 또는 교집합 연관관계)에 따라 상이한 알고리즘(단계 350, 370 또는 380)을 적용함으로써, 각 입력 파라미터의 테스트 데이터들을 적응적으로 조합하여 테스트 케이스를 생성할 수 있다.

[0083]

이하에서는, 본 발명의 실시예들에 따른 효율성을 검증하기 위한 실험 데이터가 제시된다. 실험 데이터를 통해, 소프트웨어 컴포넌트 기반 시스템의 입출력 연관관계 구조가 각각 교집합 연관관계, 독립적 연관관계, 및 모든 연관관계 일 때 본 발명에 따라 축소된 테스트 케이스 집합이 생성되는지 검증할 수 있다. 본 발명에 따

른 축소된 테스트 케이스 생성 방법의 효율성을 검증하기 위해 축소된 테스트 케이스 생성 시 시간 복잡도를 계산하고, 전체 테스트 케이스 축소율  $R_{tc}$ 를 수학적 식 4와 같이 계산한다.

[0084] [수학적 식 4]

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

[0085] [0086] 각 실험에 따라 계산된 시간 복잡도 및 전체 테스트 케이스 축소율은 기존의 Union 알고리즘 및 기존의 ParaOrder 알고리즘과 비교하여 분석된다.

[0087] **I. 실험 1(교집합 연관관계)**

[0088] 실험 1에서는, 도 1의 소프트웨어 컴포넌트 기반 시스템(100)을 테스트 대상으로 한다. 테스트 대상 시스템은 4개의 단위 컴포넌트로 구성되어 있으며, 외부 입력 파라미터 집합( $CC_{in}$ )과 외부 출력 파라미터 집합( $CC_{out}$ )은 다음과 같이 정의된다.

[0089]  $CC_{in} = EXT_{in}(C_1) \cup EXT_{in}(C_2) \cup EXT_{in}(C_3) \cup EXT_{in}(C_4)$

[0090]  $= \{x_{1,1}, x_{1,2}\} \cup \{x_{2,1}, x_{2,2}\} \cup \Phi \cup \{x_{4,1}, x_{4,2}, x_{4,3}\}$

[0091]  $= \{x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2}, x_{4,1}, x_{4,2}, x_{4,3}\},$

[0092]  $CC_{out} = EXT_{out}(C_1) \cup EXT_{out}(C_2) \cup EXT_{out}(C_3) \cup EXT_{out}(C_4)$

[0093]  $= \{y_{1,1}, y_{1,2}\} \cup \Phi \cup \{y_{3,1}, y_{3,2}\} \cup \{y_{4,1}\}$

[0094]  $= \{y_{1,1}, y_{1,2}, y_{3,1}, y_{3,2}, y_{4,1}\}$

[0095] 테스트 명세 정보를 이용하여 각 외부 입력 파라미터 별로 확장된 명세 기반 테스트 데이터 생성 기법을 통해 테스트 데이터를 생성하면, 표 2와 같은 각 외부 입력 파라미터 별 테스트 명세 정보와 테스트 데이터 집합을 얻을 수 있다. 전체 테스트 데이터 집합  $T_d$ 는 다음과 같이 각 외부 입력 파라미터의 테스트 데이터의 합집합으로 생성된다.

[0096]  $T_d = \{D(x_{1,1}), D(x_{1,2}), D(x_{2,1}), D(x_{2,2}), D(x_{4,1}), D(x_{4,2}), D(x_{4,3})\}.$

[0097] 테스트 대상 시스템에 대한 실제 입출력 연관관계에 기초하여 IORG를 생성하면 도 5a와 같다. 생성된 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.

[0098]  $F = \{f_1, f_2, f_3, f_4, f_5, f_6, f_7\}, f_1 = \{x_{1,1}, D(x_{1,1})\}, f_2 = \{x_{1,2}, D(x_{1,2})\}, f_3 = \{x_{2,1}, D(x_{2,1})\}, f_4 = \{x_{2,2}, D(x_{2,2})\}, f_5 = \{x_{4,1}, D(x_{4,1})\}, f_6 = \{x_{4,2}, D(x_{4,2})\}, f_7 = \{x_{4,3}, D(x_{4,3})\},$

[0099]  $X(y_{1,1}) = \{x_{1,1}, x_{1,2}\}, X(y_{1,2}) = \{x_{1,2}, x_{2,1}\}, X(y_{3,1}) = \{x_{2,1}, x_{2,2}\}, X(y_{3,2}) = \{x_{2,2}\}, X(y_{4,1}) = \{x_{4,1}, x_{4,2}, x_{4,3}\}.$

[0100] 여기서,  $X(y_{3,2}) \subset X(y_{3,1})$ 이 되어  $X(y_{3,2})$ 의 연관관계는 삭제되고, 결과적인 집합  $R$ 은  $R = \{r_1, r_2, r_3, r_4\}, r_1 = \{f_1, f_2\}, r_2 = \{f_2, f_3\}, r_3 = \{f_3, f_4\}, r_4 = \{f_5, f_6, f_7\}$ 이 된다. 즉, 각 연관관계 집합  $r_1, r_2, r_3$  사이에는 서로 교집합이 존재하며  $r_4$ 만 독립적인 관계가 된다.

[0101] 전체 테스트 케이스 축소율  $R_{tc}$ 를 측정하기 위하여, 각 외부 입력 파라미터의 테스트 데이터를 모두 조합(All combination)하여 다음과 같이 128개로 구성된 전체 테스트 케이스 집합을 생성한다.

[0102] 
$$T_{in} = D(x_{1,1}) \times D(x_{1,2}) \times D(x_{2,1}) \times D(x_{2,2}) \times D(x_{4,1}) \times D(x_{4,2}) \times D(x_{4,3})$$

[0103] 
$$|T_{in}| = 2 * 2 * 2 * 2 * 2 * 2 * 2 = 128.$$

[0104] 표 10은 테스트 대상 시스템의 전체 테스트 케이스 축소율을 나타낸다. 128개로 구성된 전체 테스트 케이스 집합에 대해, Union 알고리즘은 20개로 구성된 축소된 테스트 케이스 집합을 생성하여 84.37%에 해당하는 테스트 케이스 수를 감소시켰으며, 본 발명에 따른 알고리즘과 ParaOrder 알고리즘은 8개의 축소된 테스트케이스를 생성하여 93.75%에 해당하는 테스트 케이스 수를 감소시켰다. Union 알고리즘의 경우  $r_1, r_2, r_3$ , 각 연관관계에 포함된 외부 입력 파라미터에 대한 테스트 케이스를 서로 조합하여 각각 4개의 테스트 케이스를 생성하고,  $r_4$ 에 대해 8개의 테스트 케이스를 생성한다. 그리고 각 연관관계에 포함되지 않은 외부 입력 파라미터에 대해서는 랜덤하게 실제 값을 추가한 후 합하여 20개의 축소된 테스트 케이스를 생성한다. 본 발명에 따른 알고리즘과 ParaOrder 알고리즘은 "don't care data"에 대한 처리를 미리 하지 않기 때문에 8개의 테스트 케이스를 생성한다. 여기서, 각 연관관계가 교집합 연관관계를 가지므로, 본 발명에 따른 알고리즘에서는 HybridOrder 알고리즘에 의해 축소된 테스트 케이스 집합을 생성한다.

표 10

생성기법	$R_{tc}$	$ T_{in}^{red} $
Union[22]	84.37%	20
ParaOrder[24]	93.75%	8
제안한 알고리즘	93.75%	8

[0105]

[0106] 표 11은 각 알고리즘이 축소된 테스트 케이스를 생성하는데 요구되는 시간 복잡도를 나타낸다. Union 알고리즘의 경우, 각 연관관계에 속한 외부 입력 파라미터들의 테스트 데이터를 조합하는데 요구되는 시간에 의해 시간 복잡도가 계산되므로 간단한 시간 복잡도를 가진다. ParaOrder 알고리즘의 경우, 7개의 외부 입력 파라미터들을 추가하여 축소된 테스트 케이스를 생성하므로 시간 복잡도가 증가한다. 본 발명에 따른 알고리즘의 경우,  $r_4$ 에 대한 조합을 먼저 생성하고 나머지 외부 입력 파라미터들을 조합하기 때문에 ParaOrder에 비해 시간 복잡도가 감소됨을 확인할 수 있다.

표 11

생성기법	$O(.)$	비고
Union[22]	20	$O(\sum_{k=1}^4 \prod_{f_i \in r_k} D(f_i))$
ParaOrder [24]	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 알고리즘	76	$O(\prod_{f_i \in r_4} D(f_i) + \sum_{i=1}^4 ((D(f_i) \times \sum_{k=1}^m \prod_{f_j \in r_k} D(f_j) \times  T_{ci} )))$

[0107]

[0108] II. 실험 2(독립적 연관관계)

[0109] 실험 2에서는, 도 8의 소프트웨어 컴포넌트 기반 시스템(800)을 테스트 대상으로 한다. 테스트 대상 시스템은 장애물을 피해 로봇을 이동시키면서 안내 메시지를 출력하는 시스템으로서, 거리 센서 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 구성되어 있으며, 외부 입력 파라미터 집합( $CC_{in}$ )과 외부 출력 파라미터 집합( $CC_{out}$ )은 다음과 같이 정의된다.

[0110]  $CC_{in} = EXT_{in}(C_{dsc}) \cup EXT_{in}(C_{mc}) \cup EXT_{in}(C_{sgc}) \cup EXT_{in}(C_{cc})$

[0111]  $= C_{in}^{dsc} \cup \Phi \cup \{sgc\_in1\} \cup \Phi$

[0112]  $= \{dsc\_in1, dsc\_in2, dsc\_in3, dsc\_in4, dsc\_in5, dsc\_in6, dsc\_in7, dsc\_in8, sgc\_in1\}$ ,

[0113]  $CC_{out} = EXT_{out}(C_{dsc}) \cup EXT_{out}(C_{mc}) \cup EXT_{out}(C_{sgc}) \cup EXT_{out}(C_{cc})$

[0114]  $= \Phi \cup \{mc\_out1\} \cup \{sgc\_out1\} \cup \Phi$

[0115]  $= \{mc\_out1, sgc\_out1\}$ .

표 12

ID	Description	Equivalence partitions( $D(x_i)$ )	$D(x_i)$	$ D(x_i) $
dsc_in1	Forward left sensor data	$0 < x \leq 1$	0.5	4
		$1 < x \leq 2$	1.2	
		$2 < x \leq 3$	2.9	
		$3 < x$	5.0	
dsc_in2	Forward middle sensor data	$0 < x \leq 1$	0.5	4
		$1 < x \leq 2$	1.2	
		$2 < x \leq 3$	2.2	
		$3 < x$	3.1	
dsc_in3	Forward right sensor data	$0 < x \leq 1$	0.5	4
		$1 < x \leq 2$	1.2	
		$2 < x \leq 3$	2.2	
		$3 < x$	4.0	
dsc_in4	Back left sensor data	$0 < x \leq 1$	1.0	3
		$1 < x \leq 2$	1.2	
		$2 < x$	2.8	
dsc_in5	Back middle sensor data	$0 < x \leq 1$	0.3	3
		$1 < x \leq 2$	1.2	
		$2 < x$	3.4	
dsc_in6	Back right sensor data	$0 < x \leq 1$	0.9	3
		$1 < x \leq 2$	2.0	
		$2 < x$	4.2	
dsc_in7	Range boundary of forward sensors	$0 < x \leq 3$	2.0	2
		$3 < x$	3.1	
dsc_in8	Range boundary of back sensors	$0 < x \leq 2$	1.2	2
		$2 < x$	2.8	
sgc_in1	Language types of a guide message	Korean language	KOREA	2
		English language	ENGLISH	

[0116]

[0117] 테스트 명세 정보를 이용하여 각 외부 입력 파라미터 별로 확장된 명세 기반 테스트 데이터 생성 기법을 통해 테스트 데이터를 생성하면, 표 12와 같은 각 외부 입력 파라미터 별 테스트 명세 정보와 테스트 데이터 집합을 얻을 수 있다. 전체 테스트 데이터 집합  $T_d$ 는 다음과 같이 각 외부 입력 파라미터의 테스트 데이터의 합집합으로 생성된다.

[0118]  $T_d = \{D(dsc\_in1), D(dsc\_in2), D(dsc\_in3), D(dsc\_in4), D(dsc\_in5), D(dsc\_in6), D(dsc\_in7), D(dsc\_in8),$

$D(\text{sgc\_in1})\}$ .

[0119] 테스트 대상 시스템에 대한 실제 입출력 연관관계에 기초하여 IORG를 생성하면 도 9와 같다. 생성된 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.

[0120]  $F = \{f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9\}$ ,  $f_1 = \{\text{dsc\_in1}, D(\text{dsc\_in1})\}$ ,  $f_2 = \{\text{dsc\_in2}, D(\text{dsc\_in2})\}$ ,  $f_3 = \{\text{dsc\_in3}, D(\text{dsc\_in3})\}$ ,  $f_4 = \{\text{dsc\_in4}, D(\text{dsc\_in4})\}$ ,  $f_5 = \{\text{dsc\_in5}, D(\text{dsc\_in5})\}$ ,  $f_6 = \{\text{dsc\_in6}, D(\text{dsc\_in6})\}$ ,  $f_7 = \{\text{dsc\_in7}, D(\text{dsc\_in7})\}$ ,  $f_8 = \{\text{dsc\_in8}, D(\text{dsc\_in8})\}$ ,  $f_9 = \{\text{sgc\_in1}, D(\text{sgc\_in1})\}$

[0121]  $X(\text{sgc\_out1}) = \{\text{dsc\_in1}, \text{dsc\_in2}, \text{dsc\_in3}, \text{dsc\_in7}, \text{sgc\_in1}\}$ ,  $X(\text{mc\_out1}) = \{\text{dsc\_in4}, \text{dsc\_in6}, \text{dsc\_in6}, \text{dsc\_in8}\}$ .

[0122] 여기서,  $R = \{r_1, r_2\}$ ,  $r_1 = \{f_1, f_2, f_3, f_7, f_9\}$ ,  $r_2 = \{f_4, f_5, f_6, f_8\}$ 이 된다. 즉, 각 외부 입력 파라미터들간의 연관관계  $r_1 \cap r_2 = \emptyset$  이므로 서로 독립된 연관관계를 가진다.

[0123] 전체 테스트 케이스 축소율  $R_{tc}$ 를 측정하기 위하여, 각 외부 입력 파라미터의 테스트 데이터를 모두 조합(All combination)하여 다음과 같이 13824개로 구성된 전체 테스트 케이스 집합을 생성한다.

[0124]  $T_{in} = D(\text{dsc\_in1}) \times D(\text{dsc\_in2}) \times D(\text{dsc\_in3}) \times D(\text{dsc\_in4}) \times D(\text{dsc\_in5}) \times D(\text{dsc\_in6}) \times D(\text{dsc\_in7}) \times D(\text{dsc\_in8}) \times D(\text{sgc\_in1})$ ,

[0125]  $|T_{in}| = 4*4*4*3*3*3*2*2*2 = 13824$ .

[0126] 표 13은 테스트 대상 시스템의 전체 테스트 케이스 축소율을 나타낸다. 13824개로 구성된 전체 테스트 케이스 집합에 대해, Union 알고리즘은 310개 구성된 축소된 테스트 케이스 집합을 생성하여 97.63%에 해당하는 테스트 케이스 수를 감소시켰으며, 본 발명에 따른 알고리즘과 ParaOrder 알고리즘은 256개의 축소된 테스트 케이스를 생성하여 98.02%에 해당하는 테스트 케이스 수를 감소 시켰다. Union 알고리즘의 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 생성한 다음, 각 집합에 포함된 "don't care data"에 대해 랜덤하게 실제 값을 추가한 후 두 집합을 합쳐 310개의 축소된 테스트 케이스를 생성한다. 본 발명에 따른 알고리즘과 ParaOrder 알고리즘은 "don't care data"에 대한 처리를 미리 하지 않기 때문에 256개의 테스트 케이스를 생성한다. 여기서, 각 연관관계가 독립적 연관관계를 가지므로, 본 발명에 따른 알고리즘에서는 Independent Union 알고리즘에 의해 축소된 테스트 케이스 집합을 생성한다.

**표 13**

생성기법	$R_{tc}$	$ T_{in}^{red} $
Union[22]	97.63%	310
ParaOrder[24]	98.02%	256
제한한 알고리즘	98.02%	256

[0127]

[0128] 표 14는 각 알고리즘이 축소된 테스트 케이스를 생성하는데 요구되는 시간 복잡도를 나타낸다. 표 14에 나타난 바와 같이, Union 알고리즘과 본 발명에 따른 알고리즘은 동일한 시간 복잡도를 가지며, ParaOrder 알고리즘은 입력 파라미터 개수에 따라 시간 복잡도가 복잡해짐을 확인할 수 있다.

표 14

생성기법	$\mathcal{O}(\cdot)$	비고
Union[22]	310	$\mathcal{O}(\sum_{k=1}^2 \prod D(f_i))$
ParaOrder[24]	86154	$\mathcal{O}(\sum_{i=1}^6 ((D(f_i) \times \sum_{k=1}^m \prod_{f_j \in \#_k} D(f_j) \times  T_{cd} )))$
제한한 알고리즘	310	$\mathcal{O}(\sum_{k=1}^2 (\prod_{f_i \in \#_k} D(f_i)))$

[0129]

[0130]

III. 실험 3(모든 연관관계)

[0131]

실험 3에서는, 도 10의 소프트웨어 컴포넌트 기반 시스템(1000)을 테스트 대상으로 한다. 테스트 대상 시스템은 네비게이션 기능을 이용하여 로봇을 이동시키면서 안내 메시지를 출력하는 시스템으로서, 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,  $C_{lsc}$ ), 모바일 컴포넌트(Mobile component,  $C_{mc}$ ), 문장 생성 컴포넌트(Statement generation component,  $C_{sgc}$ )로 구성되어 있으며, 외부 입력 파라미터 집합( $CC_{in}$ )과 외부 출력 파라미터 집합( $CC_{out}$ )은 다음과 같이 정의된다.

[0132]

$$CC_{in} = EXT_{in}(C_{gmc}) \cup EXT_{in}(C_{ppc}) \cup EXT_{in}(C_{lsc}) \cup EXT_{in}(C_{lc}) \cup EXT_{in}(C_{nc})$$

[0133]

$$\cup EXT_{in}(C_{bsc}) \cup EXT_{in}(C_{udc}) \cup EXT_{in}(C_{oac}) \cup EXT_{in}(C_{sgc}) \cup EXT_{in}(C_{mc})$$

[0134]

$$= C_{in}^{gmc} \cup \Phi \cup C_{in}^{lsc} \cup \Phi \cup \{nc\_in1, nc\_in2\} \cup C_{in}^{bsc} \cup C_{in}^{udc} \cup \Phi$$

[0135]

$$\cup \{sgc\_in1\} \cup \Phi$$

[0136]

$$= \{gmc\_in1, lsc\_in1, nc\_in1, nc\_in2, bsc\_in1,$$

[0137]

$$udc\_in1, udc\_in2, udc\_in3, udc\_in4, sgc\_in1\},$$

[0138]

$$CC_{out} = EXT_{out}(C_{gmc}) \cup EXT_{out}(C_{ppc}) \cup EXT_{out}(C_{lsc}) \cup EXT_{out}(C_{lc}) \cup EXT_{out}(C_{nc})$$

[0139]

$$\cup EXT_{out}(C_{bsc}) \cup EXT_{out}(C_{udc}) \cup EXT_{out}(C_{oac}) \cup EXT_{out}(C_{sgc}) \cup EXT_{out}(C_{mc})$$

[0140]

$$= \Phi \cup \{sgc\_out1\} \cup \{mc\_out1\}$$

[0141]

$$= \{sgc\_out1, mc\_out1\}.$$

표 15

ID	Description	Equivalence partitions (D(x <sub>i</sub> ))	D(x <sub>i</sub> )	D(x <sub>i</sub> )
gmc_in1	Map file	valid map file	officeMap.map	1
lsc_in1	Sensor data	Source sensor data	SOURCE-VALUE	6
		Destination sensor data	DESTINATION-VALUE	
		Intermediate location sensor data	INTER-LOC-VALUE	
		Left obstacle detection sensor data	LEFT-OBS-DETECT-VALUE	
		Right obstacle detection sensor data	RIGHT-OBS-DETECT-VALUE	
		Middle obstacle detection sensor data	MIDDLE-OBS-DETECT-VALUE	
nc_in1	Source coordinate	Valid source coordinate	VALID-SRC-COORDINATE	2
		Invalid source coordinate	INVALID-SRC-COORDINATE	
nc_in2	Destination coordinate	Valid destination coordinate	VALID-DEST-COORDINATE	2
		Invalid destination coordinate	INVALID-DEST-COORDINATE	
bsc_in1	Bumper sensor	Bumper is not touched	0	2
		Bumper is touched	1	
udc_in1	Left distance sensor	0 < x ≤ 1	0.5	4
		1 < x ≤ 2	1.2	
		2 < x ≤ 3	2.9	
		3 < x	5.0	
udc_in2	Right distance sensor	0 < x ≤ 1	0.5	4
		1 < x ≤ 2	1.2	
		2 < x ≤ 3	2.2	
		3 < x	3.1	
udc_in3	Middle distance sensor	0 < x ≤ 1	0.5	4
		1 < x ≤ 2	1.2	
		2 < x ≤ 3	2.2	
		3 < x	4.0	
udc_in4	Maximum boundary	0 < x ≤ 3	2.0	2
		3 < x	3.1	
sgc_in1	Language types of a guide message	Korean language English language	KOREA ENGLISH	2

[0142]

[0143]

테스트 명세 정보를 이용하여 각 외부 입력 파라미터 별로 확장된 명세 기반 테스트 데이터 생성 기법을 통해 테스트 데이터를 생성하면, 표 1와 같은 각 외부 입력 파라미터 별 테스트 명세 정보와 테스트 데이터 집합을 얻을 수 있다. 전체 테스트 데이터 집합 T<sub>d</sub>는 다음과 같이 각 외부 입력 파라미터의 테스트 데이터의 합집합으로 생성된다.

[0144]

$$T_d = \{D(gmc\_in1), D(lsc\_in1), D(nc\_in1), D(nc\_in2), D(bsc\_in1), D(udc\_in1), D(udc\_in2), D(udc\_in3), D(udc\_in4), D(sgc\_in1)\}.$$

[0145]

테스트 대상 시스템에 대한 실제 입출력 연관관계에 기초하여 IORG를 생성하면 도 11과 같다. 생성된 IORG와 T<sub>d</sub>를 이용하여 테스트 데이터를 포함하는 외부 입력 파라미터 집합 F와 각 외부 입력 파라미터들간의 연관관계 집합 R을 생성하면 다음과 같다.

[0146]

$$F = \{f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9, f_{10}\}, f_1 = \{gmc\_in1, D(gmc\_in1)\}, f_2 = \{lsc\_in1, D(lsc\_in1)\}, f_3 = \{nc\_in1, D(nc\_in1)\}, f_4 = \{nc\_in2, D(nc\_in2)\}, f_5 = \{bsc\_in1, D(bsc\_in1)\}, f_6 = \{udc\_in1, D(udc\_in1)\}, f_7 = \{udc\_in2, D(udc\_in2)\}, f_8 = \{udc\_in3, D(udc\_in3)\}, f_9 = \{udc\_in4, D(udc\_in4)\}, f_{10} = \{sgc\_in1, D(sgc\_in1)\}$$

[0147]

$$X(sgc\_out1) = CC_{in},$$

[0148]

$$X(mc\_out1) = \{gmc\_in1, lsc\_in1, nc\_in1, nc\_in2, bsc\_in1, udc\_in1, udc\_in2, udc\_in3, udc\_in4\}.$$

[0149]

여기서, X(sgc\_out1) ⊃ X(mc\_out1)의 관계를 가지므로 R={r<sub>1</sub>}, r<sub>1</sub>={f<sub>1</sub>, f<sub>2</sub>, f<sub>3</sub>, f<sub>4</sub>, f<sub>5</sub>, f<sub>6</sub>, f<sub>7</sub>, f<sub>8</sub>, f<sub>9</sub>, f<sub>10</sub>}이 된다. 즉, 테스트 대상 시스템의 특정 외부 출력 파라미터가 모든 외부 입력 파라미터의 영향을 받으므로, 축소된 테스트 케이스 집합은 전체 테스트 케이스 집합과 동일하게 된다. 각 외부 입력 파라미터의 테스트 데이터를 모두 조합(All combination)하여 다음과 같이 12288개로 구성된 전체 테스트 케이스 집합을 생성한다.

[0150]

$$T_{in} = D(gmc\_in1) \times D(lsc\_in1) \times D(nc\_in1) \times D(nc\_in2) \times D(bsc\_in1) \times D(udc\_in1) \times D(udc\_in2) \times D(udc\_in3) \times D(udc\_in4) \times D(sgc\_in1),$$

[0151]  $|T_{in}| = 1*6*2*2*2*4*4*2*2 = 12288.$

[0152] 표 16은 테스트 대상 시스템의 전체 테스트 케이스 축소율을 나타낸다. Union 알고리즘, ParaOrder 알고리즘 및 본 발명에 따른 알고리즘 모두 축소율은 0% 이며, 그 이유는 외부 입력 파라미터들간의 연관관계 집합 R에 속한 연관관계가 1개( $r_1$ )이기 때문이다. 이 경우 각 알고리즘은 전체 테스트 케이스 집합과 동일한 테스트 케이스를 생성하게 된다. 하지만 각 알고리즘이 전체 테스트 케이스 집합을 생성하는데 요구되는 시간 복잡도는 상이하 다.

표 16

생성기법	$R_{te}$	$ T_m^{red} $
Union[22]	0.0%	12288
ParaOrder[24]	0.0%	12288
제안한 알고리즘	0.0%	12288

[0153]

[0154] 표 17은 각 알고리즘이 테스트 케이스를 생성하는데 요구되는 시간 복잡도를 나타낸다. 표 17에 나타난 바와 같이, Union 알고리즘과 본 발명에 따른 알고리즘은 시간복잡도가 동일하지만, ParaOrder 알고리즘은 입력 파라미터 10개를 각각 조합하면서 테스트 케이스 집합을 생성하므로 시간복잡도가 더 복잡함을 확인할 수 있다.

표 17

생성기법	$O(.)$	비고
Union[22]	12288	$O(\sum_{k=1}^n \prod_{f_i \in F_k} D(f_i)) = O(\prod_{f_i \in F} D(f_i))$
ParaOrder [24]	100663296	$O(\sum_{i=1}^{10} ((D(f_i) \times \sum_{k=1}^m \prod_{f_j \in R_k} D(f_j) \times  T_{in} )))$
제안한 알고리즘	12288	$O(\prod_{f_i \in F} D(f_i))$

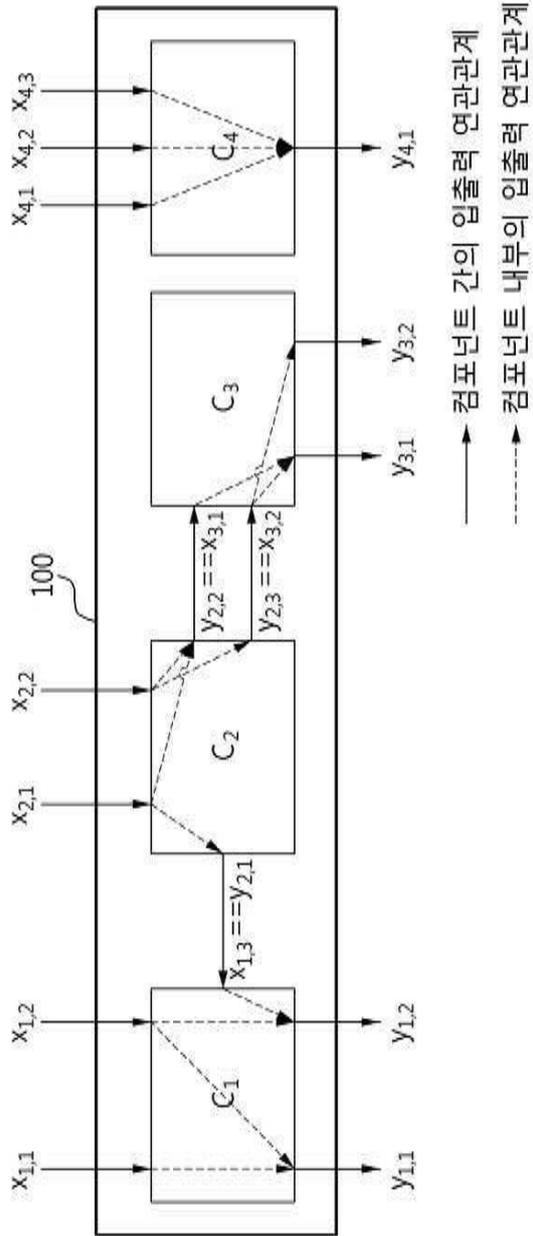
[0155]

[0156] 이상과 같이 본 발명의 실시예들이 비록 한정된 실시예와 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 상기의 기재로부터 다양한 수정 및 변형이 가능하다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 설명된 시스템, 구조, 장치, 회로 등의 구성요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다.

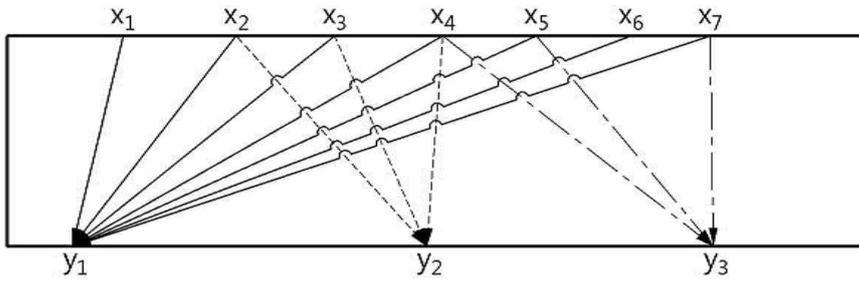
[0157] 그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.

도면

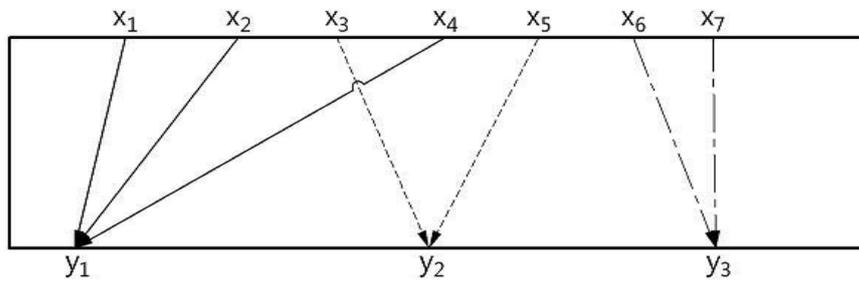
도면1



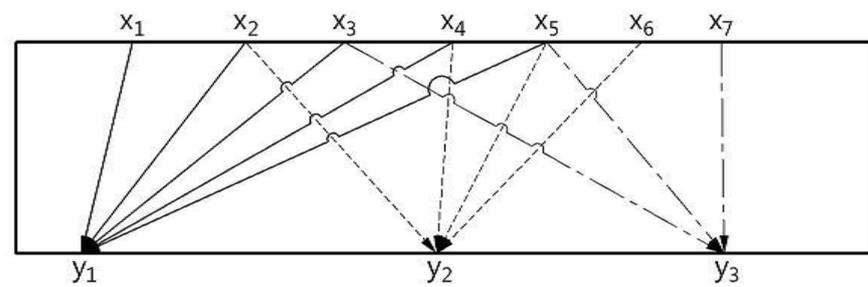
도면2a



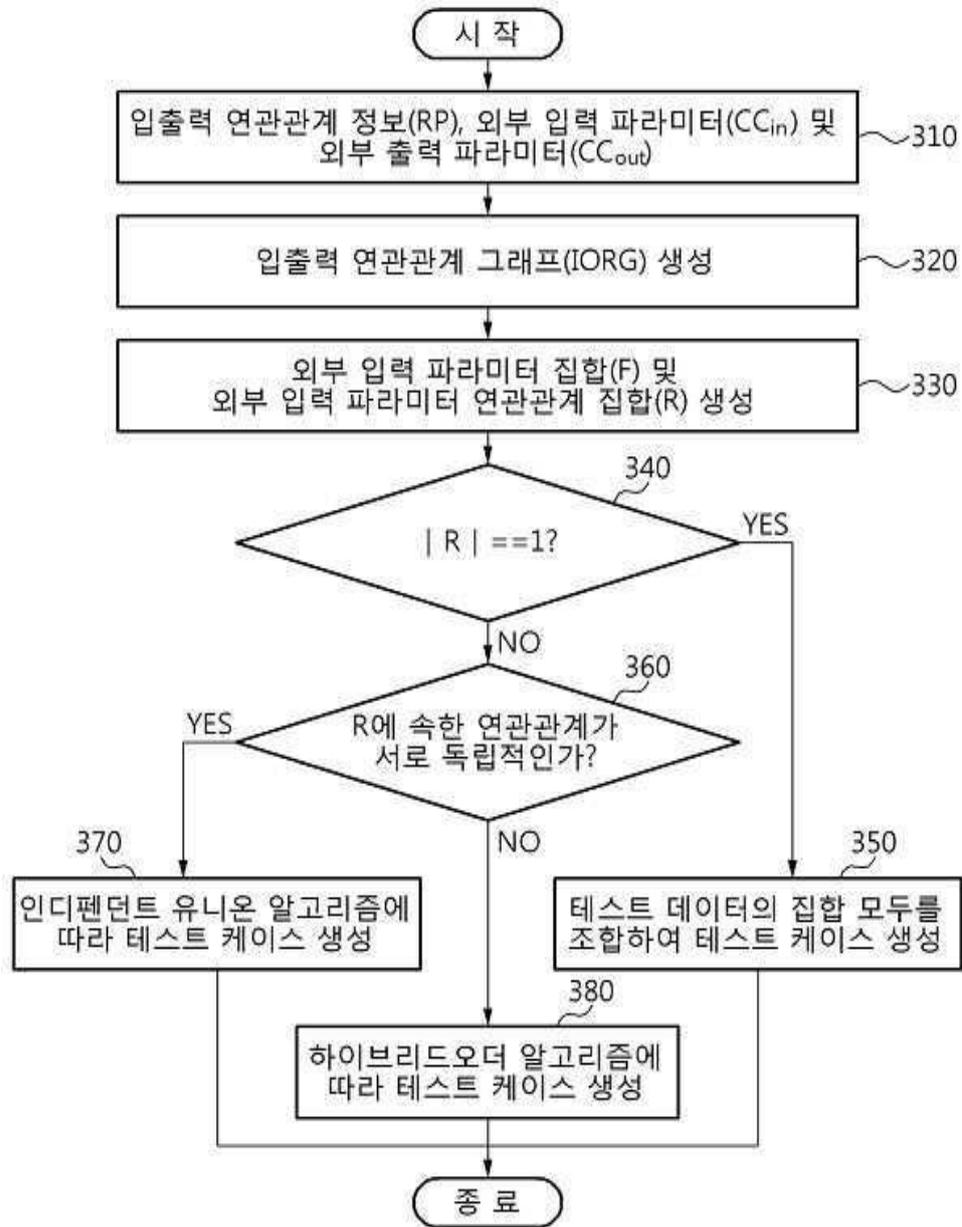
도면2b



도면2c



도면3



## 도면4

```

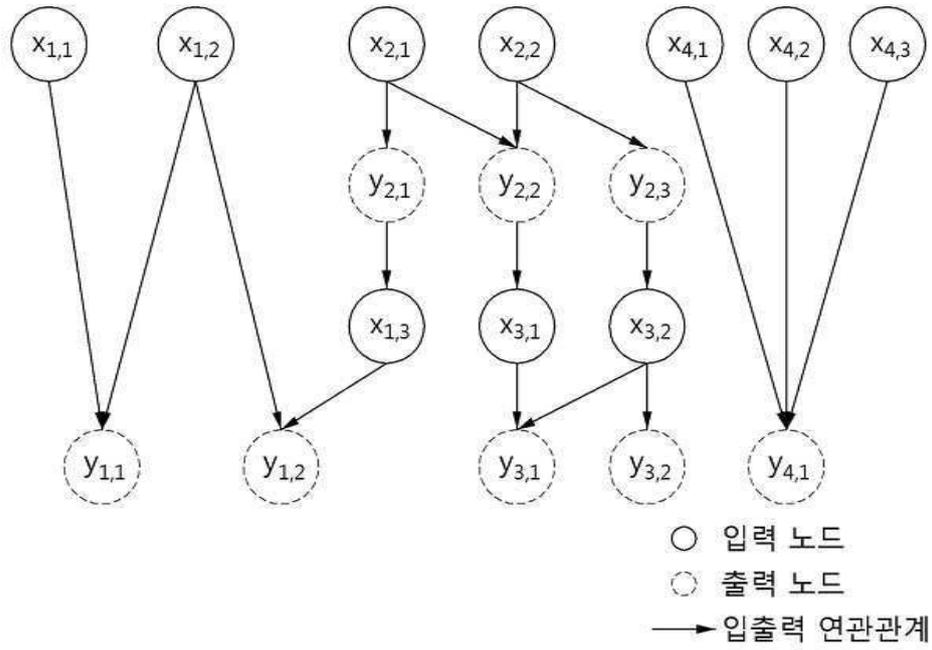
Input :  $RP$ : A set of I/O relationships of a CBS system,
           $CC_{in}$ : A set of external inputs of a CBS system,
           $CC_{out}$ : A set of external outputs of a CBS system
Output:  $IORG$ : I/O relationship directed graph

1  $IORG.V \leftarrow \emptyset$ ; /* nodes of graph IORG */
2  $IORG.E \leftarrow \emptyset$ ; /* edges of graph IORG */
3 foreach  $x \in CC_{in}$  do
4    $IORG.V \leftarrow x$ ;
5    $G \leftarrow GenerateIOReGraph(x)$ ;
6    $IORG \leftarrow IORG + G$ ;
7 end
8 return  $IORG$ ;

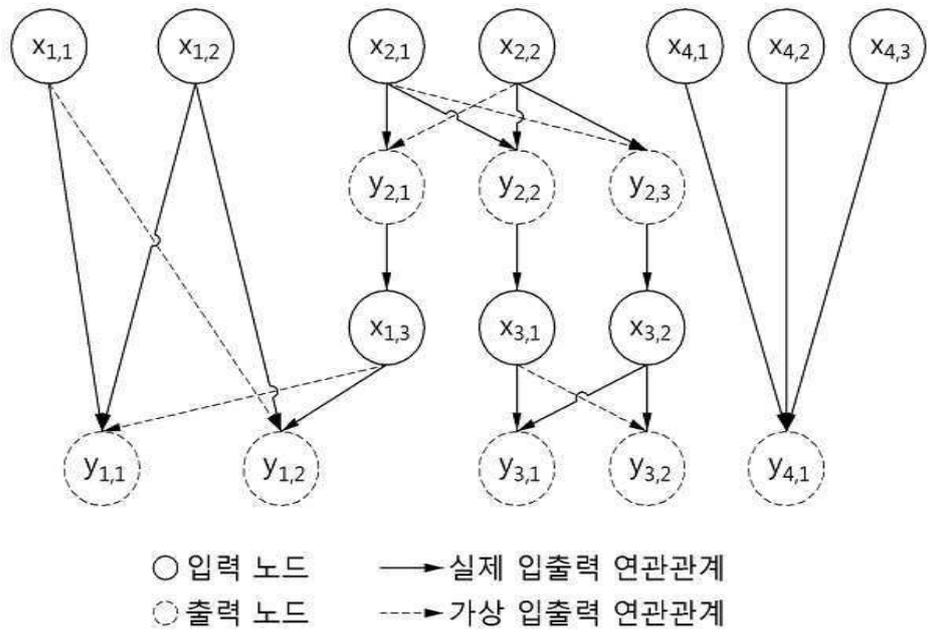
9 // algorithm as a recursive function
10 Function  $GenerateIOReGraph(inp)$  {
11   if  $RP_{intra} \neq \emptyset$  then
12      $RO \leftarrow$  Get intra-outputs relevant to  $inp$  in  $RP_{intra}$ ;
13   else // We don't know intra-component I/O
         relationships
14      $RO \leftarrow$  all outputs of the component which have  $inp$ ;
15   end
16   foreach  $rout \in RO$  do
17      $G.V \leftarrow rout$ ;
18      $G.E \leftarrow$  Add edge from  $inp$  to  $rout$ ;
19      $RI \leftarrow$  Get inter-inputs relevant to  $rout$  in  $RP_{inter}$ ;
20     if  $(RI == \emptyset) \vee (rout \in CC_{out})$  then continue;
21     foreach  $rin \in RI$  do
22        $G.V \leftarrow rin$ ;
23        $G.E \leftarrow$  Add edge from  $rout$  to  $rin$ ;
24        $NG \leftarrow GenerateIOReGraph(rin)$ ;
25        $G \leftarrow G + NG$ ;
26     end
27   end
28   return  $G$ ;
29 }

```

도면5a



도면5b



## 도면6

```

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

```

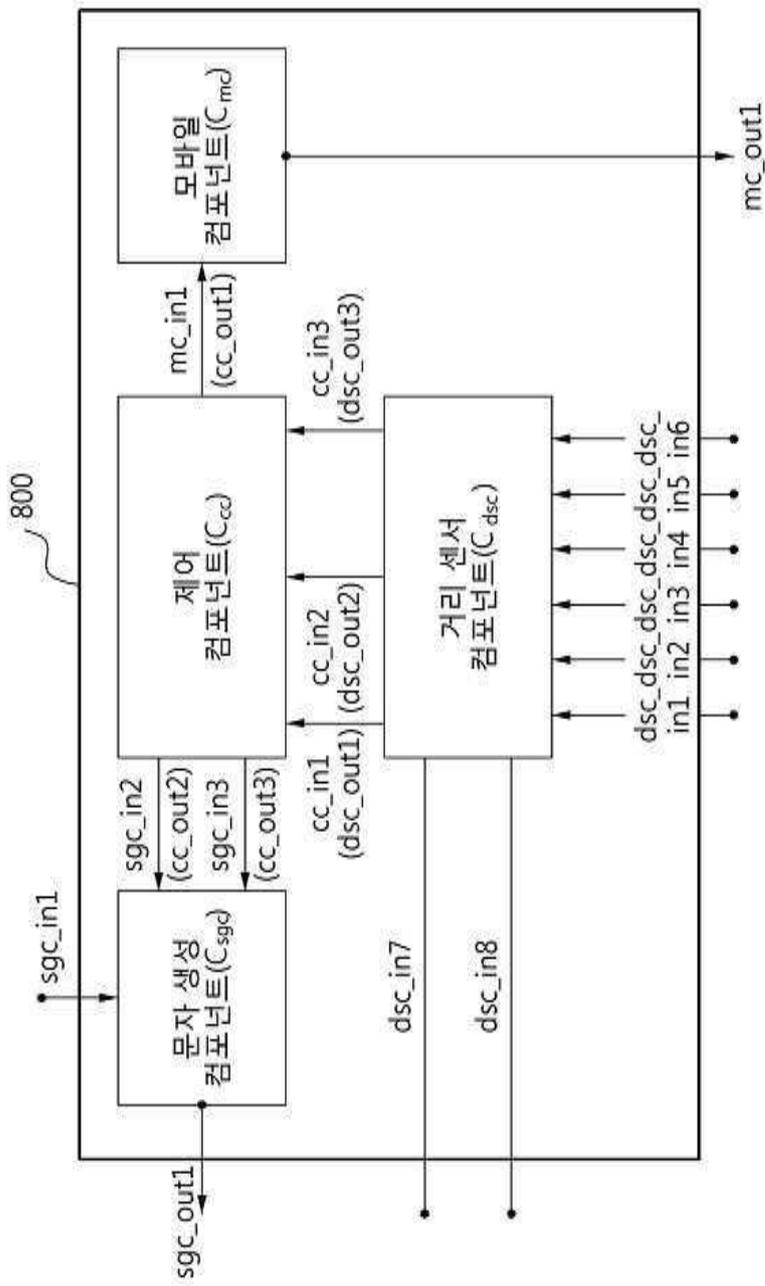
## 도면7

```

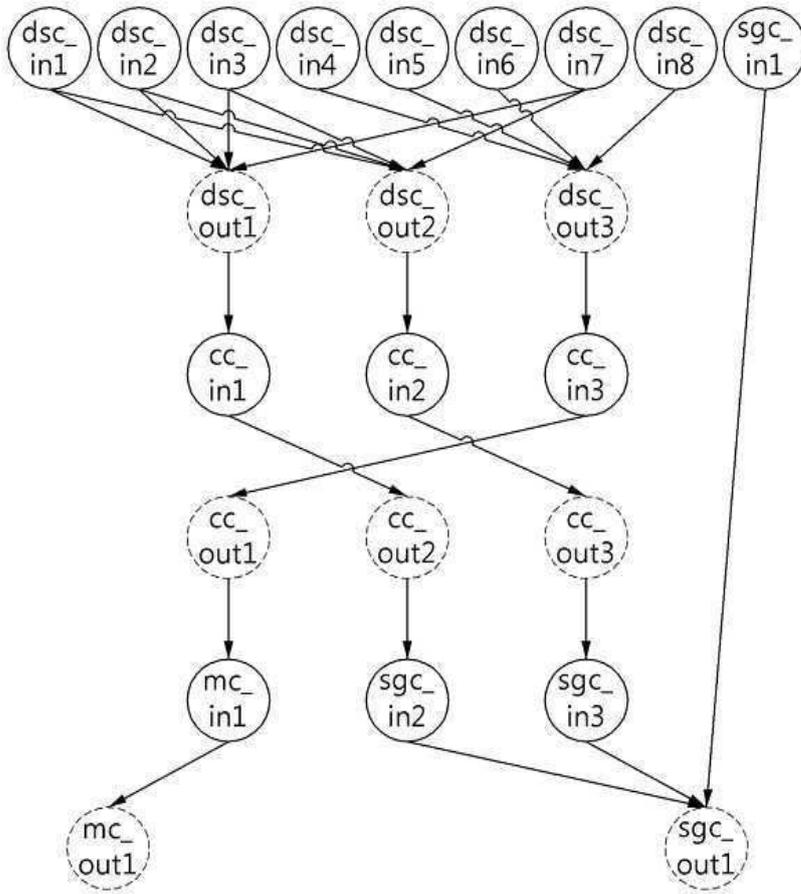
Input : F: A set of external input parameters with a set of test data
         R: A set of I/O relationships of SCBS
Output :  $T_{in}^{red}$ : A reduced set of test cases
1 //Combinations by independent relationships
2  $R_{idt} = \text{SelectIndepRelations}(R)$ ;
3 if  $R_{idt} \neq \emptyset$  then
4    $F_{idt} = \{r_k \cap F | r_k \in R_{idt}\}$ ;
5    $T_{idt} = \text{IndependentUnion}(F_{idt}, R_{idt})$ ;
6    $F = F - F_{idt}$ 
7    $R = R - R_{idt}$ 
8 end
9  $r = \text{SelectRelation}(R)$ ;
10  $size = \prod_{x_i \in r} D(x_i)$ 
11 Create size empty test cases into  $T_{in}^{red}$ ;
12 Fill all combinations of parameters in  $r$  into  $T_{in}^{red}$ ;
13  $F = F - \{f_i | f_i \in r\}$ ;
14  $F_{deal} = \{f_i | f_i \in r\}$ ;
15 // Combinations by parameters
16 while  $F \neq \emptyset$  do
17    $f_i = \text{SelectFactor}(F)$ ;
18    $Inters = \{r_k \cap (F_{deal} + f_i) | f_i \in r_k, r_k \in R\}$ ;
19    $RedInters = \text{Delete subset of element in Inters}$ ;
20    $Combset = \{(v_{ij}, \dots, v_{ij}) | v_{ij} \in V_{ij}, \dots, v_{ij} \in V_{ij}, \{f_{ij}, \dots, f_{ij}\} \in RedInters\}$ ;
21   // The follow extensions of  $T_{in}^{red}$  are similar to ParaOrder
22   Extend  $T_{in}^{red}$  horizontally;
23   Extend  $T_{in}^{red}$  vertically;
24    $F = F - \{f_i\}$ ;
25    $F_{deal} = F_{deal} + \{f_i\}$ ;
26 end
27 // Union of  $T_{idt}$  and  $T_{in}^{red}$ 
28 if  $|T_{idt}| > |T_{in}^{red}|$  then
29   // The empty test cases include "don't care data"
30   Add  $(|T_{idt}| - |T_{in}^{red}|)$  empty test cases into  $T_{in}^{red}$ ;
31 end
32 Fill test cases in  $T_{idt}$  into  $T_{in}^{red}$ ;
33 return  $T_{in}^{red}$ ;

```

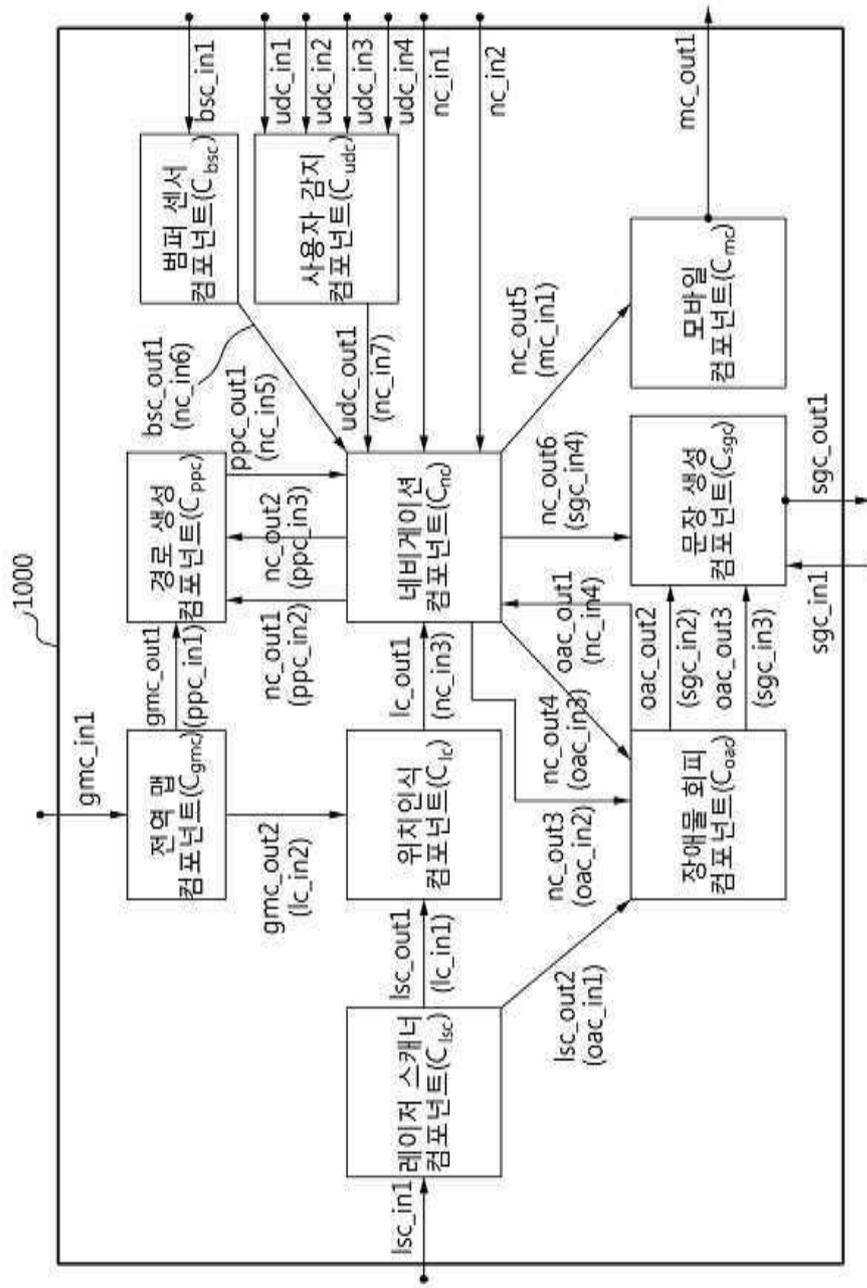
도면8



도면9



도면10



도면11

