효율적인 프로토콜 적합성 시험을 위한 변칙성 제거

이 현 철† · 허 기 택††

요 약

통신 프로토콜은 통신하는 두 개체간의 정보 교환에 관계된 법칙이라고 할 수 있다. 적합성 시험은 구현된 프로토콜이 원래의 프로토콜 규격과 일치하는가를 검사하는 것으로써 상호운용성과 비용의 효율성을 높이기 위해서 매우 중요하다. 프로토콜은 일반적으로 외부적으로 관찰 가능한 행동을 나타내는 제어흐름과 내부적으로 사용되는 변수들을 나타내는 자료흐름으로 표현할 수 있다. 지금까지 연구는 주로 프로토콜의 제어흐름만을 고려하거나, 또는 제어와 자료흐름을 상호 분리하여 처리하는 연구가 진행되어 왔다. 본 논문에서는 자료흐름의 변칙성을 분석하고, 효율적인 적합성 시험을 위해서 프로토콜에 변칙성을 제거할 수 있는 알고리즘을 제시하였다. 또한 해당 알고리즘을 실제 Estelle로된 가상 프로토콜에 적용함으로써, 자료의 변칙성이 제거됨을 보여 주었다.

Anomaly Removal for Efficient Conformance Test

Hyun-Chul Lee† · Gi-Taek Hur††

ABSTRACT

The protocol conformance testing is to check whether an implementation of a protocol conforms to its specification. And it is important to improve the interoperability of protocol and the efficiency of cost. In general, protocol is composed of the control flow representing observable behaviors and the data flow representing internally used variables. Until now, research for generation of test suite has been realized only consideration the control flow of protocol or separation control flow from data flow.

Case of considering control flow, contents of test was simple and definite. Length of test was short. But it was of little application, and it didn't manage each kind errors in data flow. Therefore, we must generate test case that can manage control and data flow simultaneously. If errors was included in protocol, anomaly of variable showed in processing data flow. So, anomaly of variable must be removed for efficient conformance testing. Therefore in this dissertation, we proposed algorithm which can remove anomaly of variable for efficient conformance testing. And it showed that anomaly of variable was got rid of applying this algorithm to real protocol.

1. 서 론

통신의 증대와 통신 구조의 복잡성이 증가함에 따라서 프로토콜의 정확성이 더욱 요구되고, 신뢰성 있는 통신 프로토콜이 필요하게 되었다. 이를 위해서 효율적으로 프로토콜을 설계하고 구현하여, 적용할 수 있도록 하기 위해서 프로토콜 설계에서 구현까지를 통합적으로 관리할 수 있어야 만 한다. 프로토콜은 일반적으로 외부적으로 관찰 가능한 행동을 나타내는 제어흐름과 입/출력뿐만 아니라 내부적으로 사용하는 변수들을 표현하는 자료흐름으로 표현할 수 있다[2].

† 정 회 원 : 동신대학교 컴퓨터학과 강사
†† 종신회원 : 동신대학교 컴퓨터학과 교수
논문접수 : 1998년 11월 9일, 심사완료 : 1998년 12월 29일

지금까지의 제시된 연구는 제어흐름에 대해 시험항목을 생성하는 기법들에 중점적으로 이루어져 왔고, 자료흐름을 부분적이나 독립적으로 다루어 왔거나 무시했다[2]. 그러나 제어흐름만을 고려한 적합성 시험은 동일 입력에 의해 다음 상태나 출력이 구분되지 않는 비결정성이 존재하고[5], 출력된 자료의 내용에 대해서는 정확성을 확인 할 수 없어, 완전한 시험 판정이 불가능하다[11].

또, 자료흐름에 존재할 수 있는 각종 오류를 처리할 수 없으므로, 제어흐름만을 고려한 적합성 시험은 많은 문제점을 내포하고 있다. 따라서 기능적으로 완전한 시험을 위해서는 프로토콜 제어와 자료흐름을 동시에 처리할 수 있는 시험항목이 생성되어야만 한다.

본 논문에서는 기능적으로 완전하고, 정확한 적합성 시험을 위해서 프로토콜이 자료흐름을 처리할 때 나타날 수 있는 변칙성을 제거할 수 있는 알고리즘을 제시하였다. 그리고 Estelle로 작성된 가상 프로토콜에 이 알고리즘을 실제 적용하여, 자료의 변칙성을 제거하였고, UIO방법을 이용하여 시험항목을 생성하였다.

2. 적합성 시험

통신프로토콜을 구현할 때 표준안에서 제시된 프로토콜에 대해 서로 다른 형태의 독립된 구현 제품들의 상호 운용성을 높이기 위한 시험 절차가 필요하다. 국제표준에 따르는 모든 프로토콜의 명세는 구현제품의 해당프로토콜에 대한 적합성 확보를 위해 필요한 적합성 요구사항을 포함하고 있다. 프로토콜의 적합성 시험은 구현된 프로토콜의 행위와 능력이 프로토콜 표준으로 정의된 요구사항에 적합하게 동작하는가를 검증하는 것으로, 관찰된 행위와 예측된 행위가 일치하는가를 검증하는 과정을 말한다. 즉, 구현 제품이 이러한 적합성 요구사항을 충족시키는 가를 검사하는 것이라고 할 수 있다. 따라서 적합성은 구현과 규격의 관계로, 구현이 규격에 대하여 맞는가를 검증하는 것이므로 따라서 규격해석방법이라고도 말할 수 있다[3,4].

구현된 제품에 대해서 적합성 시험을 하기 위해서는 시험을 위한 항목들을 미리 준비하여야 한다. 시험항목을 생성함에 있어서 중요하게 고려되어야할 사항은 시험비용의 효율성 및 오류 검출 능력등 이라 할 수 있다. 적합성 시험을 할 때 시험대상은 구현된 프로토콜의 오류를 찾기 위해 외부 정보만을 사용하는 블랙박스 시험으로 수행된다. 이 IUT의 내부행위는 직접 관찰할 수 없지만, 입력부분에서 시험항목을 적용한 후 출력부분에서 그 입력항목에 각각에 대한 출력값들을 검사한 후에 그 결과를 관찰할 수 있다.

2.1 프로토콜 제어흐름과 자료흐름

프로토콜의 제어흐름과 자료흐름에 대한 지금까지의 연구는 주로 시험경로의 선정 방법에 대해서 주로 이루어졌고, 데이터 선택에 관한 연구는 이제 시작 단계에 있다. 이제 데이터를 선택 할 때에 제어흐름과 자료흐름을 분리하여 처리하는 방법과 동시에 고려하는 방법이 있다. 제어흐름과 자료흐름을 분리해서 고려하는 경우는 제어흐름과 자료흐름을 각기 다른 그래프로 나타내 Subsequence를 생성한다. 이 경우는 각각의 연구를 분리해서 수행할 수 있다는 이점이 있는 반면 두 개의 부분에서 겹쳐지는 부분이 생길 수 있다는 단점이 있다.

제어흐름과 자료흐름을 동시에 고려하는 경우는 데이터의 논리상 존재하지 않는 경로를 생략할 수 있고, 제어흐름과 자료흐름을 모두 포함하는 보다 짧은 시험항목을 생성할 수 있다는 장점이 있다. 단점으로는 제어흐름과 자료흐름을 모두 고려해서 그래프를 만들어야 하므로, 그래프가 더 복잡해질 수 있다.

2.1.1 제어흐름

최근 적합성 시험의 길이 측면에서 가장 효율적인 시험항목 생성을 위해서는 UIO sequence 방법이 우수하다고 알려져 있다. 물론 이러한 UIO sequence 방법 외에 Transition(T) 방법, W 방법, Distinguishing Sequence(DS) 방법, RCP 방법, UIO sequence를 갖지 않을 경우에 PUIO 또는 Signature 방법, UIO방법에서 시험항목의 길이를 줄인 SUIO와 MUIO방법 등이 있다. 이와 같은 방법으로 시험항목열 생성방법에 관한 연구는 시험항목열의 길이를 줄이고, 적용가능성을 넓히며, 오류 검출범위를 확장하는 방향으로 연구가 계속되고 있다.

2.1.2 자료흐름

일반적인 소프트웨어는 외부로부터의 자료의 입력

에 의해 정해진 처리 과정을 거쳐 출력을 생성하며 내부적으로 제어흐름과 자료흐름을 갖는다. 유한상태 기계를 기초로 한 시험항목 생성방법은 프로토콜의 자료흐름 중 출력 값을 제외한 나머지 부분 즉, 상호작용 파라미터, 천이를 수행하기 위한 술어, 명세에 포함된 우선 순위나 지연, 내부변수와 같이 프로토콜에서 중요한 부분을 제외하고, 자료흐름을 부분적으로 고려하였거나 자료흐름을 무시했다. 프로토콜 적합성 시험에 있어 자료흐름은 중요한 의미를 가지고 있지만, 여기서 사용되는 변수는 그 형태와 부여되는 값이 다양하고 처리과정이 복잡하여 인간의 작업을 필요로 하는 많은 과정을 포함하고 있다. 또 사용되는 변수의 수와 범위가 커서 시험항목생성에는 많은 어려움이 따른다.

2.2 관련연구

지금까지 연구는 제어흐름에 대한 시험항목을 생성하는 기법에 중점을 두고, 자료 부분에 대한 흐름은 소홀히 되어 왔다. 그러므로 써 보다 완전한 시험열을 생성할 수 없고, 부분적으로 실행 불가능한 경우가 존재하므로 시험열의 신뢰성이 저하되고, 비결정성을 내포하는 경우 제어흐름 분석만으로 근본적인 시험이 불가능하다. 따라서 단순히 입력 값에 대한 출력 값의 결과를 관찰하는 제어흐름의 시험뿐만 아니라 프로토콜의 일부인 즉, 할당문, 조건문, 표현식 등의 구현이 바르게 되었는가에 대한 시험이 필요하게 된다.

제어흐름과 자료흐름을 고려한 선행연구를 보면, Sarikaya등은 정규화된 Estelle로부터 천이 수행을 위한 서술이나 지역, 광역 변수 등과 같은 자료흐름 처리가 가능한 혼성방법을 제안하였다[11]. 이는 정규화된 명세로부터 제어흐름과 자료흐름 그래프를 각각 분리하여 작성하고 이들을 이용하여 시험 순서를 생성하는 방법이다. 제어흐름 그래프는 Estelle의 FROM문을 원시상태로 TO문을 목적상태로 해서 이들을 연결하는 에지를 형성하면서 그래프를 생성하므로 상태의 변화를 나타내는데 이용한다. Estelle에 의한 명세는 속성을 갖는 모듈의 집합체를 기술하며 이 모듈간의 상호동작을 기술한다.

WU등은 EBE(External Behavior Expression)이라는 모델을 기초로 한 형식 프로토콜 명세로부터 시험항목을 유도할 수 있는 방법을 제시하였다[10]. EBE는 입출력 순서와 이들간의 논리적 관계를 이용하여 프로토콜 외부 행위만을 명세 한다. 이를 위한 시험항목은 순수한 시험을 위한 시험항목을 먼저 생성하고, 생성된 시험항목 중 추상적인 시험을 위한 시험항목을 선정한다. 마지막으로 수행 가능한 시험 방법을 위한 시험을 선택하는 순서로 수행된다.

Ural은 프로토콜의 제어흐름과 자료흐름을 동시에 처리할 수 있는 정적 자료흐름 분석방법을 제안하였다[6]. 이는 프로토콜의 제어 흐름과 자료흐름을 나타낼 수 있는 그래프를 각각 그리면 그 그래프 자체가 프로토콜 명세 시 정의된 모든 변수의 사용 용도를 명확하게 나타낼 수 있다. 따라서 이 정보를 이용하면 어떤 제약 조건이 부과되어도 이를 만족하는 시험항목을 생성 할 수 있게 된다.

Rapps, S등은 명세의 구성을 기준으로 하여 all-nodes, all-edges, all-defs, all-p-uses, all-c-uses/some-p-uses, all-p-uses/some-c-uses, all-uses, all-du-paths/all paths 등 시험선택의 9가지 방법을 제시하였다[11].

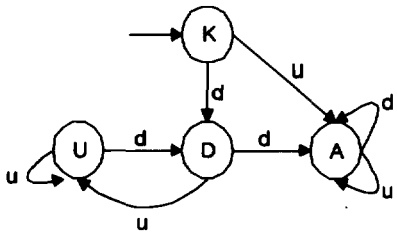
Fabureau는 제어흐름을 만들면서 동시에 자료흐름까지를 고려하여 그래프를 만드는 방법을 제안하였다[9]. 제어흐름 그래프를 만들 때 자료흐름 까지를 고려하므로 선언되지 않는 변수를 참조한다든지, 사용하지 않는 변수를 선언하는 것과 같이 논리상 존재하지 않는 결과는 고려하지 않아도 되는 이점이 있다.

3. 변칙성 제거 알고리즘

3.1 변칙성

자료흐름이란 변수의 상태와 관련된 사건들의 연속을 의미하며, 자료흐름시험에 의하여 IUT에 입력될 시험 열을 생성하고, 입력 범위 내에서 실제 입력될 특성 값을 PDU형태로 시험환경에 적용하므로써 시행된다. 일반적인 자료흐름상에서 나타나는 변칙성의 예를 들면 정의되지 않는 변수를 참조, 사용되지 않는 변수를 정의, 정의된 상태에서 재정의 등을 들 수 있다.

변칙성의 상태도를 보면 다음과 같다[2,7].



① **D** : Define된 상태
② **U** : Computational use, Predicated use된 상태
③ **K** : Killed, Undefined된 상태
④ **A** : Anomaly된 상태

(그림 1) 자료흐름의 변칙성
(Fig. 1) Anomaly of data flow

프로토콜을 기능적으로 완전한 시험을 위해서는 자료흐름의 입력 변수에 대해서 모든 변칙성을 검사해야 하나 프로토콜의 규모가 점점 커지고 복잡함으로 인해, 이러한 작업은 매우 어려운 일이며, 지나치게 많은 시간과 노력을 필요로 한다는 문제점이 있다. 프로토콜 특성상 시험열은 FDT로 기술된 프로토콜의 각 상태와 천이를 중심으로 입력된 값이 출력에 정상적으로 반영하는지를 확인하므로써 사리에 맞지 않는 자료 흐름 측면에서의 변칙성을 발견하고 또 오류를 예방하는 것을 그 목적으로 한다.

변수의 정적인 자료흐름을 기반으로 한 프로그램의 각 변수는 생성되어 소멸까지 처리과정에서 가질 수 있는 사용유형은 다음과 같이 발생한다.

① 정의(Definition or Initialized, def)
- 자료의 입력, Function Call, 할당문의 좌측에 사용되는 경우.

② 사용(Use)
- 계산적 사용(Computational Use, c-use) : 자료의 출력, 할당문의 우측에 사용된 경우.
- 조건적 사용(Predicated Use, p-use) : 조건문에 사용된 경우.

③ 도착(Reach) : 주어진 시점에 도착 가능한 정의 의 경우.

④ 생존(Live) : 임의의 블록에서 사용하여 다른 블록까지 존재한 경우.

⑤ 소멸(Killed or Undefined, K) : 생성된 정의 중에서 재 생성되었거나 정의되지 않는 경우.

⑥ 변칙적인 경우(Anomaly) : 사용되지 않는 변수를 정의, 정의된 상태에서 재정의, 무한 루프 등이 있다.

일반적인 자료의 적합성 시험은 EFSM상에서 나타나는 자료 흐름을 고려한다. 자료 흐름의 시험항목 생성은 자료 흐름의 개념을 사용하며 정적자료흐름분석을 통한 시험항목을 생성한다. 이 자료흐름의 분석의 목적은 원시코드가 주어졌을 경우 원시코드 상에서 발생할 수 있는 자료흐름 변칙성을 탐지하는 것이다.

3.2 Estelle로 명세화된 가상 프로토콜

Estelle은 유한상태기계 모델을 기초로 하고 있으며, 계산 처리를 위해 파스칼의 형식으로 확장한 프로토콜 명세 언어로, 모듈은 계층 구조를 이루며, 채널이라는 인터페이스 구조를 통하여 PDU형태의 메시지와 서비스를 주고받는 구현지향서술언어이다.

제어흐름 그래프는 Estelle의 FROM문을 원시 상태로 TO문을 목적 상태로 해서 이들을 연결하는 에지를 형성하면서 그래프를 생성하므로, 상태의 변화를 나타내는데 이용한다. Estelle에 의한 명세는 속성을 갖는 모듈의 집합체 및 모듈간의 상호동작을 기술한다.

Estelle 명세화의 경우 상호작용 파라메터를 모두 고려할 경우 함수간의 관계 인식에 지나치게 많은 노력이 요구된다. 때문에 모델링 단계에서는 Estelle명세는 단일모듈명세만을 고려한다는 전제하에 테스트스위트 생성한다. 천이조건은 다음절들 중 하나 혹은 그 이상의 절에 의해 이루어지며, 전이 조건은 다음과 같이 표시한다.

① From 절 : 현재의 제어 상태
② When 절 : 입력 상호동작
③ Provided 절 : Boolean 표현
④ To절 : 다음 제어 상태를 표시
⑤ Begin-End절 : 할당문, 출력문, 프로시져 등.

그리고 자료흐름 그래프는 변수 값을 결정하는 입력 파라미터 값이 어떻게 결정되는가를 나타내고, 그 값에 따라서 출력 파라미터 값이 정해진다. 따라서 시험열은 먼저 제어흐름 그래프를 기준으로 원하는 보조 시험열을 선정하고 자료흐름 그래프에서 원하는 시험

을 위한 자료 값을 구하는 순서로 이루어진다. 여기서 사용하는 그래프는 정점(Vertex)과 에지(edge)들의 유한집합으로 구성된 그래프의 에지에 방향이 부여된 방향성그래프(Dgraph) 이다.

3.3 변칙성 제거 알고리즘

효율적인 프로토콜을 구현하기 위해서는 자료흐름상에 포함된 각종 변칙성을 제거해야 한다. 이를 위해서 다음과 같은 변칙성 제거 알고리즘을 제시했다.

```
Algorithm
Step(1). Estelle로 된 가상 프로토콜을 FSM으로 변환함.
Step(2).
        Search Subtour ;
        while {
              if ( Variable = Anomaly ) in single transition block
                        then Variable remove ;
              if (Dgraph Condition != ok)
                        then Variable remove;
               }
Step(3).
         while {
                if ( Subtour != Valid ) remove ;
               }
Step(4). EXECUTE UIO Sequence Program ;
         /* Creat 시험항목 */
         END ;
```

(알고리즘 1) 변칙성 제거 알고리즘

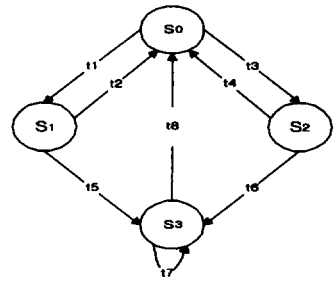
이 알고리즘은 네 단계로 구성되어진다. 단계 1에서는 Estelle로 된 가상 프로토콜을 제어흐름에 대한 유한상태기계로 작성한다. 단계 2에서는 제어흐름도에서 나타난 모든 가능한 부분경로를 파악하고, 여기에 자료흐름상의 단일전이블록 내에서 각 변수의 변칙성 여부를 조사한다. 그리고 가상 프로토콜의 변칙성을 확인하기 위해, 방향성 그래프에 의하여 만족되어야 할 조건을 적용한 다음 변칙성으로 판정된 변수는 제거한다. 그리고 단계 3에서는 제어흐름에서 나타난 부적합한 경로를 제거하고, 마지막 단계에서는 UIO Sequence 생성 프로그램을 이용하여 시험항목을 생성한다.

단계 3에서 제시한 가상프로토콜에 대해 자료흐름에 대한 변칙성을 확인하기 위해 방향성그래프로 표현할 때 만족해야 할 조건은 다음과 같다.

① V에 포함되는 모든 노드는 초기상태로부터 도달할 수 있다.
② Graph G안에는 무한루프는 존재하지 않는다.
③ V에 있는 하나의 node에 대하여 하나의 변수에 대한 두개의 선언이 그 변수에 대한 c-use 없이 존재하지 않는다.
④ 모든 def는 광역 또는 지역 def 이다.
⑤ Graph G의 변수의 모든 p-use나 Global c-use는 그 변수의 def를 포함하는 경로를 통하 여 시작노드에서 도달할 수 있다.
⑥ Graph G의 루프 없는 패스상의 변수의 선언에 대하여, 만일 그 변수가 경로 안에 포 함된다면, 최소한 하나의 c-use나 p-use가 존재한다.

3.3.1 가상프로토콜의 제어 흐름

다음 (그림 2)는 (그림 1)의 가상 프로토콜을 FSM 형태의 제어흐름 그래프로 나타낸 것으로, S_0를 초기상태로 하여 S_0, S_1, S_2, S_3의 4개의 상태와 각 상태간의 천이 즉, t_1~t_8의 8개의 전이로 구성되었음을 보여 주고 있다.



(그림 2) 가상프로토콜의 제어흐름도
(Fig. 2) Control flow of imaginary protocol

위에서 만들어진 그래프를 이용하여 부분경로를 선정하면 다음과 같다.

- $t_1 + t_2$
- $t_1 + t_5 + (t_7)^* + t_8$
- $t_3 + t_4$
- $t_3 + t_6 + (t_7)^* + t_8$

3.3.2 자료흐름 변칙성 검사

다음의 <표 1>은 각 변수에 대해 사용 유형을 분석한 자료로, 분석결과 변수별 사용 유형의 조합에 변칙성의 유무를 조사하였다.

<표 1>에서 변수 in.tpdusize, to.block, user.reason, in.buffer, out.buffer, disc.reason은 단일전이블럭내에서

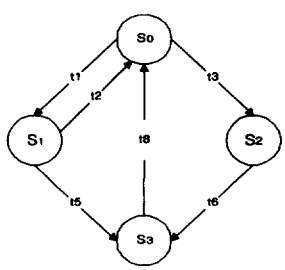
변칙성으로 나타나지만, in.tpdusize, to.block, user.reason, in.buffer, disc.reason은 사용된 블럭이 서로 다른 전역 변수가 존재하므로 가상프로토콜에 대해 자료흐름에 대한 변칙성을 확인하기 위해 방향성그래프로 표현할 때 만족해야 할 조건을 변수에 적용해보면, V에 있는 하나의 node에 대하여 하나의 변수에 대한 두개의 선언이 그 변수에 대한 c-use없이 존재하지 않는다라는 조건에서 to.block, disc.reason은 하나의 변수에 대한 두 개의 선언이 c-use를 사용했지만, in.buffer는 c-use 없이 하나의 변수에 대해 두 개의 선언이 이루어 졌으므로 변칙성이 있음을 보여주고 있다.

〈표 1〉 전이블럭내의 자료흐름 변칙성 검사
〈Table 1〉 Anomaly check of data flow in transition block

def	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	변칙성 유무
to.addr	du								
tpdu.size	du								
in.tpdusize	du	u							○
to.block		d	d		u				○
user.reason		du			du	u			○
qts.req			du			du			
local.refer				du					
source.refer				du					
in.buffer				d			d		◎
option			du	duu					
out.refer			du			uu			○
dest.refer					du				
remote.reason						du			
out.buffer				u			uu		◎
disc.reason					d			du	○

3.3.3 자료흐름 변칙성 제거

이 가상 프로토콜에 있어서 최종적인 변칙성으로 판정 된 것은 **in.buffer**와 def없이 사용된 **out.buffer**이다. 위에서 변칙성으로 판정된 변수를 포함한 부적합한 천이를 제외한 제어흐름도는 다음 (그림 3)과 같다.



(그림 3) 변칙성을 제거한 제어흐름도
(Fig. 3) Control flow removed anomalies

변칙성을 제거한 후 제어흐름에 대한 부분경로를 구하면 다음과 같고, 변칙성을 제거하기 이전의 제어흐름에 대한 부분경로에 비해 경로가 줄어들었음을 볼 수 있다.

- $t_1 + t_2$
- $t_1 + t_5 + t_8$
- $t_3 + t_6 + t_8$

위에서 구해진 부분경로만을 이용해 UIO Sequence를 구하면 다음 <표 2>와 같다.

〈표 2〉 Subtour에 대한 UIO 열
〈Table 2〉 UIO sequence for a subtour

State	UIO
0	(to.addr/in.tpdu.size)
1	(to.block/user.reason),(user.reason/dest.refer)
2	(qts.req/out.buffer, disc.reason/disc.reason)
3	(disc.reason/disc.reason)

3.4 적용결과 및 분석

본 논문은 정확한 적합성 시험을 위해서, 프로토콜 변칙성을 제거할 수 있는 알고리즘을 제시하였다. Estelle로된 가상 프로토콜을 작성하여, 이 프로토콜에 변칙성 제거 알고리즘을 적용하여 단일 블록 내에서 바로 변칙성을 검사하고, Estelle로 가상 프로토콜을 자료흐름에 포함된 변칙성을 제거하기 위하여 방향성 그래프로 표현할 때 만족해야 할 조건을 적용해 최종 변칙성으로 나타난 변수 및 경로를 제거하였다.

〈표 3〉 UIO 순서 비교
〈Table 3〉 Compare with UIO sequence

State	변칙성 제거 전	변칙성 제거 후
0	(to.addr/in.tpdu.size)	(to.addr/in.tpdu.size)
1	(to.block/user.reason), (user.reason/dest.refer)	(to.block/user.reason), (user.reason/dest.refer)
2	(option/sourse.refer)	(qts.req/out.buffer, disc.reason/disc.reason)
3	(in.tpdusize/out.buffer), (disc.reason/disc.reason)	(disc.reason/disc.reason)

변칙성이 제거되기 전의 UIO Sequence와 제거된 후의 UIO Sequence의 시험 순서를 다음 <표 3>에서

비교하였다. 상태3에서 변칙성이 제거 되기 전의 UIO Sequence는 (in.tpdusize / out.buffer)와 (disc.reason / disc.reason)의 두 개의 시험 항목이 있지만, 변칙성 제거후 UIO Sequence는 하나의 시험 항목 (disc.reason / disc.reason)만이 존재한다.

본 논문에서는 두 단계의 변칙성 검사로, 자료흐름에서 변칙성으로 판정된 변수가 제거됨으로, 분석 대상의 수가 줄어들어 개발자의 개입을 최소화하였다. 또 변칙성 검사 후, 제어 흐름에서 나타난 실행 불가능한 경로를 제거하였으므로, 제어흐름과 자료흐름으로 통합된 경로에는 부적합한 경로가 존재하지 않는다. 따라서 시험항목 생성 시 발생 될 수 있는 복잡성을 어느 정도 단순화하였고, 시험 대상이 되는 경로는 축소되어, 논리적으로 명확해진다. 그리고 제어흐름만을 고려한 시험보다는 보다 효과적이고, 신뢰성 있으며, 적은 비용의 시험 열을 생성 할 수 있다.

4. 결 론

통신에 대한 사용자의 요구 사항이 다양해짐에 따라 프로토콜도 점점 복잡해지고 있다. 따라서 구현된 프로토콜이 명세서의 요구 사항을 충족하는지를 독립적으로 시험하는 작업인 적합성 시험을 위한 연구의 필요성이 점점 더 증대해 가고 있다. 적합성 시험에 사용되는 시험항목의 타당성은 시험환경 및 적용방법과 함께 중요하다. 지금까지 통신프로토콜에 대한 연구는 자료흐름을 무시하고, 제어흐름만을 분석하였다. 그러나 자료흐름을 고려하지 않을 경우 완전한 시험이 불가능하고, 생성된 시험항목에 부분적으로 실행 불가능한 경우가 포함되는 등 많은 문제점을 내포하게 된다.

본 논문에서는 프로토콜에 대한 표준 명세로부터 시험항목을 생성하기까지 기존 연구에 대한 전반적인 사항 및 적용방안에 대해 기술하였다. 그리고 기존에 이루어 졌던 적합성 시험에 대해, 제어흐름과 자료흐름을 고려한 시험항목생성을 가상프로토콜을 작성하여, 자료흐름에 대한 변칙성 검사 후 변칙성으로 판단된 부적합한 경로를 제거하고, 이를 제어흐름에 통합했다. 이를 기본으로 변수의 정적인 측면에서 UIO방법을 이용하여 시험항목을 생성하고 활용 하는 측면에서 이루어 졌다.

앞으로의 연구 과제는 구조적인 연구를 바탕으로 기능적이고, 동적인 분석과정을 병행하여 자동화도구를 생성하는 방법과 내부변수전체의 자료흐름 및 모듈간의 통신을 고려한 연구가 진행되어야 할 것이다.

참 고 문 헌

[1] 김광현, "적합성 시험을 위한 시험항목 생성 및 오류발견 능력분석", 광운대학교 박사 학위논문, 1996.

[2] 오병호, 이상호, "자료 흐름의 변칙성을 고려한 테스트 스위트 생성기법", 정보과학회논문지, 제23권, 제10호, pp.1041-1048, 1996.10.

[3] 이상호, S.T.Vuong, "프로토콜 적합성시험을 위한 통합 환경", 정보과학회논문지, 제21권, 제5호, pp. 944-960, 1994.

[4] 이재용 외, "적합성 시험용검증언어 연구 최종 연구 보고서", 한국전자통신연구소, 1990.

[5] 허기택, "비결정성 제거에 근거한 통신프로토콜의 효율적인 시험항목 생성", 광운대학교 박사학위논문, 1994.

[6] 허기택, 이동호, "효율적인 프로토콜 시험을 위한 비결정성 제거 알고리즘", 한국통신학회논문지, 제18권, 10호, pp.1752-1581, 1993.

[7] B.Beizer, Software Testing Technique, Van Nostrand Reinhold, pp.145-172, 1990.

[8] B.Sankaya and G.V.Bochmann, E.Cerny, "A Test Design Methodology for Protocol Testing," IEEE Tran. on SE, Vol.SE-13, No.5, pp.518-531, 1987.

[9] Fareau,J.P., Hogrefe, D. and Kroon, J., *"Formal Methods in Conformance Testing : Status and Expectations,"* Working Draft Reports, 1992.

[10] Hasan Ural, *"Test Sequence Selection Based on Static Data Flow Analysis,"* Computer Communication 10(5), October, pp.234-242, 1987.

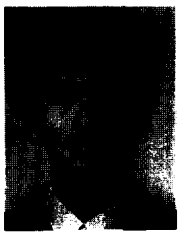
[11] Rapps, S and Weyujer, E. J., *"Data flow analysis techniques for test data selection,"* Technical Report 023 Dept. of Computer Science, Courtant Institute of Math. Science, New York University(1981).

[12] Wu, J. P. and Chanson, S. T., *"Test Sequence*

Derivation Basedon External Behavior Expression," 2nd International Workshop on Protocol Test Systems, October, pp.1-16, 1989.

[부록] Estelle로 명세화된 가상 프로토콜

```
when tcreq(to.addr)
from s0
provided to.addr = ok
to s1
t1: begin
tpdu.size:= . . .;
in.tpdu.size:= tpdu.size;
output(in.tpdu.size)
end;
when tcreq(to.block)
from s1
provided in.tpdu.size < > ok
to s0
t2: begin
user.reason:=...;
output disp(user.reason);
end;
when tcres(qts.req)
from s0
provided qts.req ≥ nil
to s2
t3: begin
option:=...;
to.block:= . . .;
out.refer:=option;
output disp(out.refer);
end;
when cr(option)
from s2
provided option < > ok
to s0
t4: begin
local.refer:= option;
source.refer:= local.refer;
in.buffer:= out.buffer;
output cc(source.refer);
end;
when tcreq(user.reason)
from s1
provided
to s3
t5: begin
disc.reason:= to.block;
dest.refer:= user.reason;
output disp(dest.refer);
end;
when tcres(qts.req)
from s2
provided qts.req > user.reason
to s3
t6: begin
remote.reason:= . . . ;
out.refer:= remote.reason;
output disp(out.refer);
end;
when data
from s3
provided out.buffer < > nil
to s3
t7: begin
in.buffer:=...;
remove(out.buffer);
end;
when ndind(disc.reason)
from s3
provided
to s0
t8: begin
output tdind(disc.reason);
end;
```



이 현 철

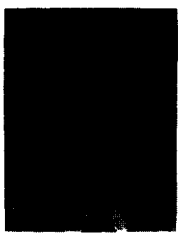
e-mail : hclee@dongshinu.ac.kr

1996년 2월 동신대학교 전자계산학과 졸업(이학사)

1998년 3월 동신대학교 대학원 전산통계학과 졸업(이학사)

1998년 3월~현재 동신대학교 컴퓨터학과 시간강사

관심분야 : 멀티미디어, 프로토콜 공학, 보안



허 기 택

e-mail : gthur@dongshinu.ac.kr

1984년 2월 전남대학교 계산통계학과 졸업(이학사)

1986년 2월 전남대학교 대학원 계산통계학과 졸업(이학석사)

1994년 2월 광운대학교 전자계산학과 졸업(이학박사)

1989년 3월~현재 동신대학교 컴퓨터학과 부교수

1990년 3월~현재 동신대학교 정보전산센터 소장

관심분야 : 멀티미디어 컨텐트분야, 프로토콜 공학