

# 비 최소 다중 UIO를 이용한 최적화된 시험열 생성

正會員 임 상 용\*, 양 대 헌\*, 강 명 호\*, 송 주 석\*

## Optimal Test Sequence Generation Scheme Using Non-minimal Multiple UIO

Sang Yong Lim\*, Dae Hun Nyang\*, Myung Ho Kang\*, Joo Seok Song\* *Regular Members*

### 요 약

본 논문에서는 최소 다중 UIO를 사용했을 때보다 비 최소 다중 UIO를 사용했을 때 시험열의 길이가 줄어들 수 있음을 보여준다. 본 논문에서 제안하는 시험열 생성 방법은 Shen의 방법이 어떤 상태에서 들어오는 에지와 나가는 에지의 차가 클 때에는 최적화된 시험열을 생성하지 못한다는 데에 기초하고 있다. 이 성질을 이용하여 시험열의 길이를 줄이는 관점에서 UIO에 기초한 보다 향상된 기법을 제안하고 기존의 방법과 시험열의 길이와 적합성 검사를 위하여 걸리는 시간, 시험열을 생성하는데 걸리는 시간이 보다 효율적임을 기존의 방법과 비교하여 설명한다.

### ABSTRACT

In this paper, we show that the length of the test sequence can be reduced using non-minimal multiple UIO instead of minimal multiple UIO. Our test sequence generation method starts from the observation that Shen's approach cannot optimize the length of the test sequence when the difference between the number of incoming edge and that of outgoing edge is large. Using the property, we propose a more efficient method to generate a UIO-based test sequence in terms of the length, and compares our method with other existing methods in terms of execution time for conformance testing and time for generation of test sequence.

### I. 서 론

프로토콜을 구현 할 때 구현물이 실제 명세와 같은가를 비교하기 위한 방법을 적합성 시험(Conformance Testing) 방법이라고 부른다[8]. 실제로 최근 몇년 동안

여러가지 적합성 시험열 생성(Conformance Test Sequence Generation) 방법이 제시 되었는데, 이런 방법들은 모두 프로토콜 명세를 FSM(Finite State Machine)으로 모델링하여 적합성 시험열을 생성한다. 여러 프로토콜 적합성 시험 방법중에 널리 쓰이는 방법들로 DS(Distinguishing Sequence), W(Characterization Set), UIO(Unique Input Output), PDS(Pairwise Distinguishing Sequence)등이 있다[1, 2, 3, 4].

\*연세대학교 컴퓨터과학과  
論文番號:97292-0819  
接受日字:1997年 8月 19日

프로토콜 구현물 자체는 블랙 박스(Black Box)이기 때문에 구현자는 가능한 모든 입력에 대하여 올바른 결과가 출력되는지 검사한다. 만약 모든 입력에 대하여 올바른 결과가 출력된다면 구현자는 구현된 프로토콜이 제대로 구현되었다고 말할 수 있다. 하지만 아주 간단한 프로토콜 자체도 많은 입력, 출력이 존재할 수 있으므로 시험열 생성과 검사는 상당히 어려워질 수 있다. 프로토콜 시험열 생성 방법은 크게 두 가지로 나뉘어 지고 있다. 첫번째는 구현이 잘못되지 않았음을 보장하기 위한 오류 보정(Fault tolerant)에 중점을 두는 방법이고[5, 9], 두 번째는 시험열의 길이를 최적화하는 데 중점을 두는 방법이다[1, 3, 7, 10]. 본 논문에서 제시하는 방법은 시험열의 길이를 최적화 하는 데 중점을 두고 있다.

시험열의 길이를 최적화 하기 위한 여러가지 방법이 제안되고 있는데, 가장 간단한 방법으로는 최소 UIO(Minimal UIO)와 RCPT(Rural Chinese Postman Tour)를 사용하여 시험열을 생성하는 방법들이 있다[1, 6]. 이 방법은 간단하다는 장점이 있지만 나중에 적합성 시험을 하기 위하여 시험열 생성시 RCPT를 구성하기 위한 에지의 중복이 많이 일어나게 된다. 따라서 전체 시험열의 길이가 길어지게 된다. 이를 개선한 방법으로 최소 다중 UIO(Minimal Multiple UIO)를 이용해서 RCPT를 구성하는 방법이 있는데, 각 상태에는 한 개 이상의 최소 UIO가 존재하고 여러개의 최소 UIO를 사용하면 시험열 생성을 위한  $G'$  그래프 생성시 더 짧은 길이의 RCPT가 존재할 확률이 높아질 수 있다는 데에 기초하고 있다. 본 논문에서 제안하는 방법도 다중 UIO에 기초하고 있다. 하지만 최소 다중 UIO가 항상 최적화된 시험열을 생성하지는 않는다. 어떤 FSM은  $m$  최소 다중 UIO를 사용했을 때, 보다 최적화된 시험열을 생성한다. 본 논문에서는 이런 예제를 제시하고, 이를 해결할 수 있는 알고리즘을 제시한다.

## II. 적합성 시험을 위한 그래프 이론

통신 프로토콜은 통신하는 대상간의 정보를 교환하는 규칙들을 나타내는 것으로 여러 국내 표준화 기구에서 표준으로 정의하고 있다. 그러나 이를 구현하는 구현자 나름대로 표준을 해석하여 구현하게 되므로

많은 해석상의 오류 및 구현 전차상의 오류-예를 들면, 입력 오류, 출력 오류, 트랜지션 오류-로 인하여 실제 구현물은 규격대로 구현되지 않을 수 있다. 따라서 정확하게 구현되었는지를 검증하는 절차를 수행하게 되는데 이를 적합성 시험(Conformance Testing)이라고 한다. 구현물 자체는 블랙 박스이기 때문에 일반적으로 화이트 박스(White Box)를 검사하는 방법과는 다르게 검사되어야 한다. 이 장에서는 본 논문에서 제안하는 적합성 시험열 생성 방법을 위한 그래프와 FSM에 관한 용어를 정의한다.

FSM은  $G = \{V, E\}$ 와 같은 형태의 방향성 그래프(Directed Graph)로 나타낼 수 있다. 여기서  $V = \{v_1, v_2, \dots, v_n\}$ 는 FSM에서의 상태이고, 방향성 에지( $v_i, v_j$ ;  $L$ ),  $L = a_k/o_i$ 은  $s_i$  상태에서 입력  $a_k$ 를 받고  $s_j$  상태로 바뀌면서 출력  $o_i$ 를 내 보낸다.

프로토콜을 검사하기 위해서는 하나의 에지를 검사하기 위해서 생성되는 시험열을 모아서 전체 시험열을 생성한 후 이 시험열에서 오류가 발생하는 지를 검사한다. 시험열을 만드는 데는 여러가지 방법이 있는데, 한 방법은 초기 입력(reset input)과 초기 상태(initial state)에서 검사하고자 하는 상태로 이동하는 최단 거리(shortest path) 입력열을 이용해서 각 트랜지션이 올바른 출력을 내는지 검사하고, 전이한 상태가 올바른 상태인가를 확인하면서 모든 트랜지션을 방문하여 시험열을 생성하는 방법이다[4]. 다른 방법으로 트랜지션 방문 방법이 있는데 오류 검사 능력이 다른 방법에 비해 떨어진다[3]. 시험열 생성시에는 테스트 세트먼트의 중복이 일어나지 않도록 하므로써 시험열의 길이를 줄일 수 있다.

정의 1 ( $G'$  그래프의 비용) 본 논문에서 최소 다중 UIO와  $m$  최소 다중 UIO를 사용했을때의 성능을 분석하기 위하여 비용의 정의를 다음과 같이 내렸다.

$$C(G') = \text{중복된 에지 } E \text{의 개수}$$

지금까지 적합성 시험이란 무엇이며,  $G'$  그래프의 비용에 대해 알아 보았다. 여기서 설명되지 않은 기호는 [1]에 나타나 있다. 다음 장에서는 기존의 시험열 생성 방법과 본 논문에서 제시하는 시험열 생성 방법을 비교하여 설명한다.

### III. 시험열 생성

#### 3.1 최소 다중 UIO를 사용한 시험열 생성

프로토콜 구현물을 검사하기 위해서는 이 프로토콜을 검사할 수 있는 시험열을 생성해야 한다. 프로토콜은 FSM으로 기술될 수 있고, 기술된 FSM을 가지고  $G'$  그래프를 생성하여, 만약  $\Delta(G')$ 의 값이 0이 된다면 이 때 시험열의 길이는 최적화 된다. 일반적으로 최소 UIO를 사용했을 때는 각 상태에 대하여 하나의 UIO만 사용하므로  $\Delta(G')$  값이 0이 되지 않는다.

각 상태에는 적어도 한 개 이상의 최소 UIO가 있다. 이런 여러개의 UIO를 이용하면 각 상태로 들어오는 여러개의 트랜지션에 대하여 서로 다른 UIO를 사용하여  $E_C$ 를 여러개의 상태로 분산시킨다. 따라서  $G'$  그래프 생성시 비용  $\Delta(G')$ 의 값이 0이 될 확률이 높아진다. 또한 이때 생성되는 시험열의 길이도 최적화 된다.

그림 1은 최소 다중 UIO를 설명하기 위한 예제 FSM이다. 그림 1을 가지고 각 상태에서 최소 다중 UIO와 최소 다중 UIO를 적용한 후 전이하는 상태는 표 1과 같다.

또한, 각 상태에서의 최소 다중 UIO를 사용하여 UIO를 할당하는  $G_M$  그래프를 구하면 그림 2와 같다.

이 경우에  $G_M$  그래프를  $E_C$ 만으로 RCPT 할 수 없다. 따라서 최소 비용으로 E를 중복하기 위하여 그림

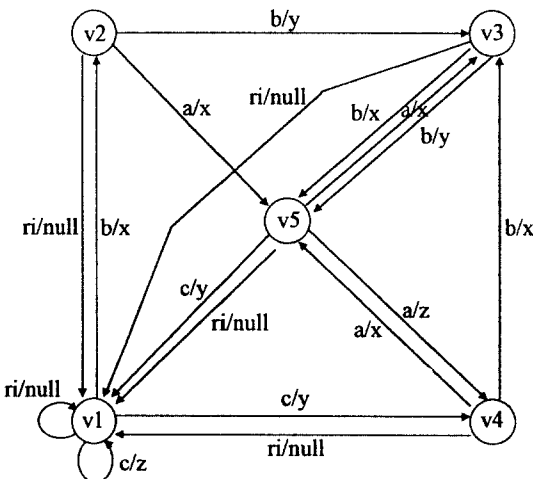


그림 1. 예제 FSM

표 1. 각 상태의 최소 다중 UIO와 TAIL(UIO<sub>i</sub>)

현재 상태	최소 다중 UIO	TAIL(UIO <sub>i</sub> )
v1	c/z	v1
v2	b/y b/x, b/y b/y	v5
v3	b/y c/y, b/y a/z	v1, v4
v4	b/x b/x	v5
v5	a/z	v4

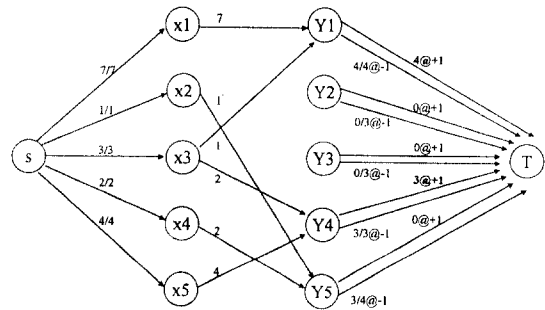


그림 2. 최소 다중 UIO 할당 그래프  $G_M$

3과 같은 Min/Max Flow 그래프를 그려서  $G'$  그래프에 E를 중복한다. 중복되는 예지는  $v_1$  상태에서  $v_2$  상태로 3개,  $v_1$  상태에서  $v_4$  상태로 2개,  $v_4$  상태에서  $v_3$  상태로 3개,  $v_4$  상태에서  $v_5$ 로 1개이다. 점선으로 표시된 것이 중복된 E이고 그 위의 숫자는 중복된 개수이다.

그림 3에서 보듯이 여기서 중복되는 E의 개수는 9개이다. 따라서 최소 다중 UIO를 사용했을 때의 비용  $C(G')$ 은 9이다.

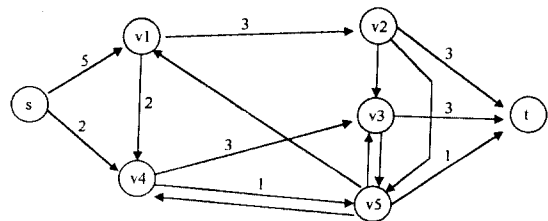


그림 3. 최소 다중 UIO를 사용한 Min/Max Flow 그래프  $G_F$

Min/Max Flow 그래프에서 얻어진 결과를 가지고, 최소 다중 UIO를 사용하여  $G'$  그래프를 구성하면 그림 4와 같이 된다. 만약 모든 상태에서 이 인덱스의 값이 0이라면 에지 중복(edge replication)을 위한 Rural Symmetric Augmentation이 필요없게 되고, 따라서 이때의 시험열의 길이는 최적화 된다.

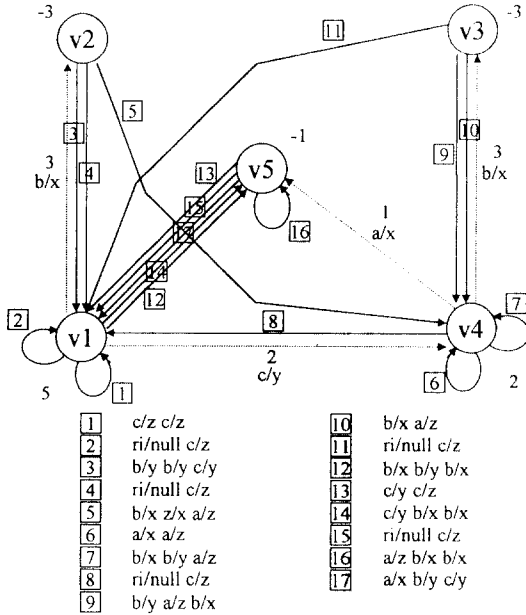


그림 4. 최소 다중 UIO를 사용한 rural symmetric augmentation 그래프  $G'$

여기서 RCPT를 이용하여 얻어지는 시험열은 표 2와 같게 되고 시험열의 총길이는 50이다. 생성 순서는 그림 4에서의 순서와 같다.

표 2. 최소 다중 UIO를 사용하여 생성된 시험열

$ri/null$	$c/z$	$c/z$	$ri/null$	$c/z$	$b/x$
$b/y$	$b/y$	$c/y$	$b/x$	$ri/null$	$c/z$
$a/x$	$a/z$	$a/x$	$a/z$	$b/x$	$b/y$
$a/z$	$b/x$	$b/y$	$a/z$	$b/x$	$a/z$
$ri/null$	$c/z$	$b/x$	$b/y$	$b/x$	$c/y$
$c/y$	$b/x$	$b/x$	$ri/null$	$c/z$	$c/y$
$a/z$	$b/x$	$b/x$	$a/x$	$b/y$	$c/y$

3.2 비 최소 다중 UIO를 사용한 시험열 생성  
최소 다중 UIO를 가지고  $G'$  그래프 생성시에  $G_M$

그래프의  $(Y, T)$  부분에서 비용이 +1인 부분에 흐름이 있게 되면  $\Delta(G')$ 의 값이 0이 되지 않는다. 따라서  $G'$  그래프를 RCPT하기 위하여 Min/Max Flow 그래프를 그려 최소 비용으로 E를 중복한다. 하지만 그림 4에서 보듯이 Min/Max Flow 그래프 생성시 소스 상태에서 소스 상태로, 또는 싱크쪽의 상태에서 싱크쪽의 상태로 흐름이 생기게 되면 이런 흐름은 시험열을 최적화 시키지 못한다. 이것으로 인해 전체 시험열의 길이를 증가시키게 된다. 본 논문에서 제시하는 방법은 UIO 할당 그래프에서 UIO의 길이가 하나 더 긴 비 최소 다중 UIO를 이용하여  $G_M$  그래프 생성시  $(Y, T)$  부분에서 비용이 +1인 부분에 흐름이 없게 하여 시험열의 길이가 최적화되게 한다. 또한 이렇게 생성되는 시험열은 최소 다중 UIO를 이용한 시험열보다 항상 그 길이가 적거나 같게 된다.

일반적으로 최소 다중 UIO는 길이가 길지 않으므로(보통 5개 미만) UIO를 찾는데 있어서 시간이 오래 걸리지 않는다. BFS(Breadth First Search)와 같은 최소 다중 UIO를 구하는 알고리즘을 이용하면 길이가 하나 더 긴 비 최소 다중 UIO를 찾는 데도 유사한 시간내에 구할 수 있다.

그림 1을 가지고 각 상태에서 비 최소 다중 UIO와 비 최소 다중 UIO를 적용한 후 전이하는 상태는 표 3과 같다.

표 3. 각 상태의 비 최소 다중 UIO와  $TAIL(UIO_j)$

현재 상태	비 최소 다중 UIO	$TAIL(UIO_j)$
v1	$c/z, c/z b/x, c/z c/y, c/y a/x$	v1, v2, v4, v5
v2	$b/y b/x, b/y b/y$	v5
v3	$b/y c/y, b/y a/z$	v1, v4
v4	$b/x b/x$	v5
v5	$a/z, a/z b/x, a/x b/x, a/x b/y$	v3, v4, v5

그림 2에서 나타나는 것처럼  $v_1$  상태에서는 최소 다중 UIO가 1개 뿐이어서,  $v_1$  상태로 들어오는 모든 에지에 대하여 하나의 UIO를 할당하게 된다. 하지만  $Y_1$  상태에서는 4개만 받아들일 수 있다. 이로 인하여  $G_M$  그래프 생성시  $(Y_1, T)$ 의 비용이 +1인 부분에 4개의 흐름이 생기게 되었다. 마찬가지로  $v_5$  상태도 최소 다중 UIO가 1개 뿐이어서  $(Y_4, T)$ 의 비용이 +1인

부분에서 3개의 흐름이 생기게 된다. 따라서 rural symmetric augmentation을 하기 위해서는 E의 중복이 필요하게 되고, 이를 위해서 Min/Max Flow 그래프를 그렸을때 그림 4에서 보듯이  $v_1$  상태에서  $v_4$  상태로 2개의 흐름이 생기게 되었다. 이런 흐름은 시험열의 길이를 최적화되지 못하게 만든다.

$v_1$  상태와  $v_5$  상태를 길이가 하나 더 긴 비 최소 다중 UIO를 사용하여 ( $Y_j, T$ )의 비용이 +1 부분에 흐

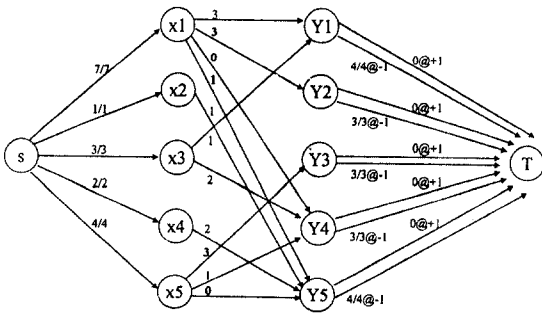


그림 5. 비 최소 다중 UIO 할당 그래프  $G_M$

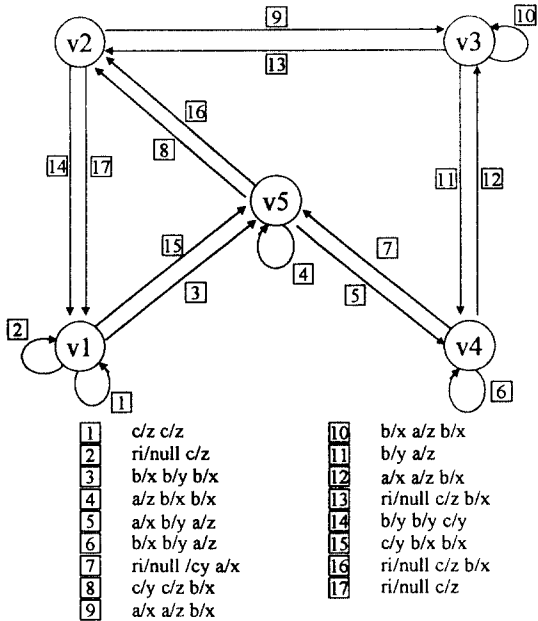


그림 6. 비 최소 다중 UIO를 사용한 rural symmetric augmentation 그래프  $G'$

표 4. 비 최소 다중 UIO를 사용하여 생성된 시험열

$ri/null$	$c/z$	$c/z$	$ri/null$	$c/z$
$b/x$	$b/y$	$b/x$	$a/z$	$b/x$
$a/x$	$b/y$	$a/z$	$b/x$	$a/z$
$b/x$	$b/x$	$a/z$	$ri/null$	$c/y$
$a/x$	$a/z$	$b/x$	$b/x$	$a/z$
$b/x$	$b/x$	$a/z$	$b/x$	$b/y$
$a/x$	$a/z$	$b/x$	$ri/null$	$c/z$
$b/x$	$ri/null$	$c/z$	$b/x$	$b/y$
$b/y$	$b/y$	$c/y$	$c/y$	$b/x$
$ri/null$	$c/z$	$b/x$	$ri/null$	$c/z$

름이 없도록 UIO를 할당하면 그림 5에서 보듯이 비용  $C(G)$ 는 비 최소 다중 UIO를 만들기 위해 추가된 7이다. 앞의 최소 다중 UIO를 사용한 방법에 비하면 2개가 줄었다. 볼드체로 표시된 숫자는 비 최소 다중 UIO를 할당하는 트랜지션의 개수이다.

비 최소 다중 UIO를 사용하여 구성한  $G'$  그래프는 그림 6과 같고, 비 최소 다중 UIO를 사용하여 얻어지는 시험열은 표 4와 같게 되고 시험열의 총 길이는 48이다.

#### IV. 시험열 생성 알고리즘

3.2절에서 살펴본 것처럼 최소 다중 UIO를 이용하는 것 보다 비 최소 다중 UIO를 이용하는 것이 생성되는 시험열의 길이를 더 짧게 할 수 있다. 이 절에서는 주어진 FSM에 대하여 비 최소 다중 UIO를 사용하여 시험열을 생성하는 방법과 이렇게 해서 생성된 시험열은 항상 최소 다중 UIO를 이용한 시험열보다 짧은 길이를 갖는다는 것을 증명한다.

비 최소 다중 UIO를 사용하여 시험열을 생성하는 방법은 다음과 같다:

1. 각 상태에 대한 최소 다중 UIO를 찾는다.
2. step 1)의 UIO를 이용하여 UIO 할당 그래프  $G_M$ 을 얻는다.
  - 2.1 각  $X_i$ 의 최소 다중 UIO에 대하여 도착상태  $Y_j$ 로의 에지를 그린다.
  - 2.2 각  $X_i$ 에 대하여 전체 비용이 최소가 되게 만드는  $UIO_i^*$ 를 찾는다.
  - 2.3 각  $X_i$ 의 들어오는 에지에 대하여 흐름 F만큼  $UIO_i^*$ 를 할당한다.
3. 만약 step 2)에서 비용이 +1인 쪽에 흐름이 있다면 길이가 하나 더 긴 비 최소 다중 UIO를 사용하여 비용이 +1인 쪽에 흐름을 없애도록 한다.
  - 4.1 step 2)의  $G_M$  그래프에서 비용이 +1인 쪽에

흐름이 생기는 싱크쪽의 상태( $Y_j$ )를 찾는다.

4.2 각  $Y_j$ 에 대해서 흐름을 생기게 하는 소스쪽의 상태( $X_i$ )를 찾는다.

4.3 각  $X_i$ 에 대해서 비 최소 다중 UIO를 찾는다.

4.4 4.3에서 찾아진 비 최소 다중 UIO의 출발 상태와 도착상태로 이루어진 에지를  $G_M$  그래프에 추가한다.

4.5 4.4에서 얻어진  $G_M$ 에 대하여 Min/Max Flow 알고리즘을 수행하여 ( $X_i, Y_j$ )에 새로운 흐름을 할당한다.

4.6 4.5에서 얻어진  $G_M$  그래프를 가지고 rural symmetric augmentation 그래프  $G'$ 을 그리고, 이 그래프를 RCPT 하면서 시험열을 생성한다.

위의 알고리즘에 의해 생성된 UIO는 다음의 정리에 의해 항상 최소 다중 UIO보다 좋은 성능을 갖는다.

정리 1. 비 최소 다중 UIO를 이용한 시험열은 다음의 관계를 만족한다.

|비 최소 다중 UIO를 사용한 시험열|

≤ |최소 다중 UIO를 사용한 시험열|

(증명)  $G$  그래프에서 각 상태로 들어오는 에지의 합은 각 상태에서 나가는 에지의 합과 같다:  $\sum d_{in}^E(v_i) = \sum d_{out}^E(v_i)$ . 이것의 증명은 [1]에 나타나 있다. 또한 각 상태로 들어오는 에지의 합은 검사되어야 할 에지의 개수와 같다. ( $Y_j, T$ )의 비용이 +1인 쪽의 흐름의 개수는 ( $S, X_i$ )의 전체 흐름에서 ( $Y_j, T$ )의 비용이 -1인 쪽의 흐름을 뺀 차이다:

$$F(y_j, t)^+ = \sum_{(x_i, y_j) \in E^+} F(x_i, y_j) - F(y_j, t)^-. \text{ 즉, } G' \text{ 그래프를 대칭으로 만들어 RCPT가 존재하기 위해서는 } F(Y_j, T)^+ \text{ 만큼의 E가 중복 되어야 한다. 최소 다중 UIO를 사용했을 때 Min/Max Flow 그래프에서 소스쪽의 상태인 } v_i \text{에서 } v_k \text{로, 또는 싱크쪽의 상태인 } v_j \text{에서 } v_l \text{로 흐름이 생기면 필요 추가분인 } F(Y_j, T)^+ \text{ 개수보다 많은 에지가 중복되게 된다. 이와 같은 흐름이 없을 때 추가되는 에지는 최소한의 개수인 } F(Y_j, T)^+ \text{ 이다. 하지만 비 최소 다중 UIO를 사용했을 때는 각 상태에서의 UIO의 크기를 하나만 확장하여 } (Y_j, T)^+$$

쪽의 흐름을 ( $Y_j, T$ ) 쪽으로 흐르게 하여 0으로 만드므로 추가되는 에지는 항상 최소한의 개수인  $F(Y_j, T)^+$  이다. □

쪽의 흐름을 ( $Y_j, T$ ) 쪽으로 흐르게 하여 0으로 만드므로 추가되는 에지는 항상 최소한의 개수인  $F(Y_j, T)^+$  이다. □

## V. 결론 및 향후 연구 방향

본 논문에서는 비 최소 다중 UIO를 사용하여 시험열의 길이를 최적화 함으로써 프로토콜 적합성 시험에 걸리는 시간을 줄이는 데 중점을 두었다.

모든 FSM에는 하나 이상의 최소 다중 UIO가 존재하고, 따라서 길이가 하나 더 긴 비 최소 다중 UIO는 최소 다중 UIO를 사용했을 때보다도 짧게 된다. 각 상태에 대한 UIO가 많아 지기 때문에 주어진 FSM에 대하여  $G'$  그래프를 대칭으로 만들 가능성이 더 높아졌고 따라서 시험열의 길이가 최적화 될 확률 또한 높아지게 되었다. 시험열의 생성은 RCPT와 같은 알고리즘을 사용할 수 있다.

적합성 검사를 위한 시험열은  $G'$  그래프를 RCPT하여 생성한다. 만약  $\Delta(G')$ 의 값이 0이 되면 시험열의 길이가 최적화 되지만 0이 되지 않으면 RCPT를 하기 위해서 E가 중복되어야 한다. 하지만 Min/Max Flow 그래프를 사용하여 E를 중복하는 것보다 UIO 할당 그래프  $G_M$ 에서 비 최소 다중 UIO를 사용하여  $\Delta(G')$ 의 값을 0으로 만드는 것이 RCPT하기 위하여 Min/Max Flow 알고리즘을 수행하지 않아도 되고, 시험열의 길이도 최적화 되므로 시간과 비용 측면에서 보다 효율적이다.

본 논문에서는 4%의 성능 향상이 있었지만 FSM이 복잡하게 되고, 최소 다중 UIO의 개수가 작을수록,  $\xi(v_i)$ 의 값이 클수록 보다 좋은 성능 향상이 있을 것이다.

비 최소 다중 UIO를 사용하였는데도 비용이 +1인 안쪽에 흐름이 생기는 경우에, 일반적으로 비용이 +1인 쪽에 흐름이 적으면 RCPT하기 위하여 중복되는 E의 개수도 적게 된다. 하지만 어떤 경우에는 Min/Max Flow 그래프에서 중복되는 E의 개수가 최소 다중 UIO를 사용했을 때보다도 짧게 된다.  $\Delta(G')$ 이 0이 되지 않는 경우에도  $G_M$  그래프를 가지고 비용이 적은 것부터 순차적으로 증가시키면서 Min/Max Flow 그래프를 그려보면 가장 최적화된 시험열을 찾을 수 있지만 시간이 오래 걸린다. 따라서 현재 사용되고 있는 시험열 생성 방법들과 비슷한 시간 복잡도를 갖

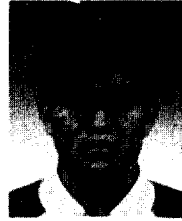
는 알고리즘에 관한 연구가 필요하다.

**참 고 문 헌**

1. Y.-N. Shen, F. Lombardi, and A.T. Dahbura, "Protocol Conformance Testing Using Multiple UIO Sequence," IEEE Transactions on comm., vol. 40, no. 8, Aug., 1992, pp. 1282-1287.
2. Krishan SABNANI and Anton DAHBURA, "A Protocol Test Generation Procedure," Computer Networks and ISDN Systems, vol. 15, 1988, pp. 285-297.
3. Alfred V. Aho, T. Dahbura, David Lee, M. Umit Uyar, "An Optimization Technique for Protocol Conformance Test Generation Based on UIO Sequences and Rural Chinese Postman Tours," IEEE Transaction on comm., vol. 39, no. 11, Nov. 1991, pp. 1604-1615.
4. Deepinder P. Sidhu, Ting-Kau Leung, "Formal Methods for Protocol Testing: A Detailed Study," IEEE Transactions on SE, vol. 15, no. 4, Apr, 1989, pp. 413-426.
5. Kshinrasagar Naik, "Fault-tolerant UIO Sequences in Finite State Machines," IWPTS'95, 1995, pp. 207-220.
6. F. Lombardi and Y.-N. Shen, "Evaluation and Improvement Fault Coverage of Conformance Testing by UIO Sequences," IEEE Trans. on Commun., vol. 40, no. 8, Aug., 1992, pp. 1288-1293.
7. Hasan Ural and Kequin Zhu, "Optimal Length Test Sequence Generation Using Distinguishing Sequences," IEEE Trans. on Networking, vol. 1, no. 3, Jun., 1993, pp. 358-371.
8. D. Rayner, "OSI Conformance Testing," Computer Networks and ISDN Systems, vol. 14, 1987, pp. 79-98.
9. Raymond E. Miller, Fellow, IEEE, and Sanjoy Paul, Member, IEEE, "Structural Analysis of Protocol Specifications and Generation of Maximal Fault Coverage Conformance Test Sequences," IEEE/ACM Trans. on Networking, vol. 2, no. 5,

Oct, 1994, pp. 457-470.

10. T. Ramalingam, K. Thulasiraman, Anindya Das, "A Generalization of the Multiple UIO Method of Test Sequence Selection for Protocols Represented in FSM," IWPTS VII, Nov. 8-10, pp. 181-196.



**임 상 용(Sang Yong Lim)** 정회원  
 1972년생  
 1992년 3월~1996년 2월: 연세대학교 컴퓨터과학과 (이학사)  
 1996년 3월~현재: 연세대학교 컴퓨터과학과(석사과정)

※주관심분야: 프로토콜 엔지니어링, 정보 보안, ATM Congestion control

**양 대 헌(Dae Hun Nyang)** 정회원  
 1970년생  
 1994년 2월: 한국과학기술대학 전기 및 전자공학과 졸업(공학사)  
 1996년 2월: 연세대학교 컴퓨터과학과 졸업(이학석사)  
 1996년 3월~현재: 연세대학교 컴퓨터과학과 박사과정 재학중

※주관심분야: 프로토콜 엔지니어링, Cryptography, ATM Congestion control

**강 명 호(Myung Ho Kang)** 정회원  
 1970년생  
 1994년 2월: 연세대학교 컴퓨터과학과 졸업(이학사)  
 1996년 2월: 연세대학교 컴퓨터과학과 졸업(이학석사)  
 1996년 3월~현재: 연세대학교 컴퓨터과학과 박사과정 재학중

※주관심분야: 정보 보안, ATM Congestion control

**송 주 석(Joo Seok Song)** 정회원  
 1953년생  
 1976년 2월: 서울대학교 전기공학과 졸업(공학사)  
 1979년 2월: 한국과학원 전기 및 전자공학과 졸업(공학 석사)  
 1979년 2월~1982년 2월: 한국전자통신 연구소 전임연구원  
 1988년 8월: Univ of California at Berkeley 전산과학과 졸업(박사)  
 1988년 9월~1989년 2월: Naval Postgraduate School Information System Department 조교수  
 1989년 3월~현재: 연세대학교 컴퓨터과학과 교수