

Estelle로부터 자료흐름이 고려된 테스트 케이스 생성

이현정, 우성희**, 이상호*
충북대학교 컴퓨터과학과*, 청주과학대학 전자계산과**

Generation of Test Cases with Data Flow From Estelle

Hyun Jeong Lee, Sung Hee Woo**, Sang Ho Lee*

Department of Computer Science, Chungbuk National University*
Department of Computer Science, Chongju Science National College**

요 약

CPN(Colored Petri Net)은 시스템을 분석하고 설계하는 모델링 기법인 페트리넷의 확장형으로서, 토큰에 color를 부여하여 자료흐름까지 고려할 수 있다.

이 연구에서는 Estelle로 기술된 프로토콜 명세를 Design/CPN에 적용가능한 CPN으로의 변환 방법을 제안하였다. 그리고 변환된 CPN을 Design/CPN에 입력하여 얻은 OG(Occurrence Graph)와 각 노드에 대한 내부 정보인 descriptor를 이용하여, 자료흐름이 고려된 보다 간편하고 효율적인 테스트 케이스를 생성하였다.

1. 서론

적합성 시험은 개발된 시스템이 OSI 표준 즉, 프로토콜 표준을 따르는지에 대한 여부를 시험하는 것으로 구현제품인 IUT에 입력/출력인 데이터 시퀀스를 적용하는 것이다. 기존의 적합성 시험은, 프로토콜 명세 언어인 SDL이나 Estelle로 기술된 프로토콜을 FSM으로 변환한 후, 개별적인 알고리즘을 사용하여 UIO 또는 서브투어(subtour)를 생성하는 복잡한 과정을 통해 이루어져 왔다. 이러한 과정을 수작업으로 진행하는 것은 비효율적이며 오류의 가능성을 내포하기도 한다.

이 논문에서는 시스템 시뮬레이션 툴로서, 시스템의 설계, 명세 분석을 위한 도구인 Design/CPN을 이용하여 프로토콜의 적합성 시험을 수행하였다. Estelle로 기술된 프로토콜 명세를 셸프루프와 부분적인 자료흐름을 고려하여 Design/CPN에 입력 가능한 CPN으로 구성하는 방법을 제안하여, [1]의 알고리즘을 이용하여 테스트 케이스를 추출하였다.

[1]에서는 Design/CPN에 적용 시 셸프루프(self-loop)를 고려하지 않았으나, 이 논문에서는 실행불가능 경로가 제거되도록 셸프루프에 순서를 부여하여 고려하였으며[5], Estelle의 PROVIDED절을 CPN의 플레이스(place)에 나타냄으로써 부분적인 자료흐름을 고려하였다.

이 논문은, 2절에서 Estelle와 Design/CPN에 대한 간략한 소개를 하였고, 3절에서 Estelle로 기술된 프로토콜 명세의 CPN 변환방법, 4절에서 사례 연구로서 제안된 방법을 Estelle로 기술된 TPO(OSI Class 0 Transport Protocol)에 적용하여 CPN으로 변환하고, 이로부터 자동적으로 생성된 내부정보로부터 UIO와 서브투어의 생성과정과 결과 분석을 하였다. 마지막으로 5절에서는 결론 및 향후 연구방향을 제시하였다.

2. Estelle와 Design/CPN

Estelle는 분산 환경에서 동시성을 요구하는 정보처리 시스템

을 기술하기 위하여 ISO에서 개발, 표준화된 FDT(Formal Description Techniques) 중의 하나로서, 특히 ISO에서 제정하고 있는 통신 프로토콜 및 서비스의 표현에 적용할 목적으로 만들어졌다. Estelle는 PASCAL에 기반을 둔 구문(syntax)과 의미(semantic)를 가지고 있고, EFSM(Extended FSM)의 정의를 쉽게 할 수 있는 등 여러 기능을 가진 언어이다. 특히 Estelle 구문은 프로토콜의 기능(function)을 표시하는 모듈(module)이라는 개념과 모듈들간의 통신을 제공해 주는 채널(channel)이라는 개념을 이용하여 구성되므로 통신 프로토콜을 기술할 때 다른 FDT 언어보다 융통성, 용이성 및 효율성이 높아 널리 이용되고 있다.

페트리 넷은 시스템을 분석하고 설계하는 모델링 기법으로서, 오브젝트와 정보의 흐름을 다루는 복잡한 분산시스템의 설계와 분석을 용이하게 한다[2].

Design/CPN은 CPN의 사용을 지원하는 툴 패키지로서, 복잡한 데이터 타입 및 복잡한 데이터 처리를 하는 CPN 모델을 지원, 잘 정의된 인터페이스를 가지는 분리된 모듈, 즉 계층적 CPN을 지원한다.

이 논문에서는 Estelle를 셸프루프와 자료흐름을 고려한 CPN 표기로 변환하여 테스트 케이스를 추출하였다.

3. Estelle 명세의 CPN 변환

3.1 Estelle의 가정 사항

Estelle는 내부 동작과 데이터 표현을 PASCAL 구문인 communicating state automata의 구조로 기술한다. 따라서 Estelle 모듈의 내부적인 동적 행위(behavior)는 CPN의 transition으로 변환된다. 변환하기 전의 Estelle 모듈에 대한 가정은 다음과 같다[3].

- (1) 모듈은 유한 개의 상태집합이다 : C_0, C_1, \dots, C_n
- (2) C_0 는 초기화 부분에서 정의된 초기상태이다.

(3) 전이 선언은 2개의 부분으로 구성되고 환경과 모듈과의 상호 동작 뿐 아니라 내부적 행위를 서술한다.

- 전이 조건
- 전이 가능할 때 다음 상태로 점화 가능함을 결정한다.
 - "FROM C" : C는 control state
 - "WHEN P(I)" : P는 입력 interaction, I는 interaction point
 - "PROVIDED B" : B는 부울 표현
- 전이 동작
- 점화 가능한 전이 중 하나의 동작을 수행하고 다음 상태에 도달하게 된다.
 - "TO D" : D는 control state
 - 전이블럭(TB): "BEGIN"과 "END" 사이의 PASCAL 구문

(4) Estelle로 기술된 프로토콜 명세는 정규화(normalize)된 형태로 입력된다.

3.2 CPN의 가정 사항

CPN은 페트리 넷에 color의 개념을 추가하여 다음과 같이 6개의 튜플(P, T, C, I, O, μ)로 구성된다.

$$CPN = (P, T, C, I, O, \mu)$$

여기서

- P : 플레이스의 집합 = {p₁, p₂, ..., p_n}, n ≥ 0
- T : 트랜지션의 집합 = {t₁, t₂, ..., t_m}, m ≥ 0
- C : color 함수, C(p) = Σk'c, k ≥ 0이고 c는 color
- I : 입력함수
- O : 출력함수
- μ(P → {0, 1, ...}) : 토큰의 초기 상태 집합
- P ∩ T = ∅ and P ∪ T ≠ ∅

3.3 Estelle의 CPN 변환

Estelle로 표현된 프로토콜 명세의 CPN 변환방법 및 대응관계는 다음과 같다.

- P = CUDU(P · B)
- T = {t_{c,k} | c ∈ C and k ∈ (P · B)}
- C(p) = C(q), if (p ∈ C & p doesn't have self-loop)
- C(r), if (p ∈ C & p has self-loop)
- C(pb), if pb ∈ (P · B)
- I(t_{c, p · B}) = {C, (P · B)}
- O(t_{c, p · B}) = D
- μ(p) = C(0) ∪ Σ(P · B)
- C(0) : 초기상태 플레이스

다음과 같은 Estelle 명세를 CPN 표기로 변환하면 그림 1과 같다.

```

WHEN P(I)
FROM C
PROVIDED B
TO D
t1 : BEGIN
.
.
END;
```

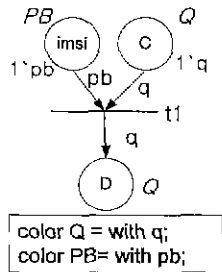


그림 1. Estelle의 CPN 변환 예

Estelle로 기술된 프로토콜은 위와

같이 정규화된 형태에서 CPN으로 변환된다. CPN은 플레이스와 트랜지션, color 선언부로 구성된다. 여기서 플레이스는 입력/조건인 카테션 프리덕트(product)로 표현된 입력/조건 플레이스와 현재 상태와 다음 상태를 나타내는 일반적인 상태 플레이스, 셀프루프를 가진 상태 플레이스의 세 종류가 있다. [1]에서는 CPN으로 변환 시 셀프루프를 고려하지 않았으나, 이 연구에서는 실행가능 경로를 갖지 않도록 셀프루프에 순서를 부여하여 CPN으로 변환하였다. 실행가능 경로는 연구 [5]에 나타나 있다.

Estelle에서의 WHEN 절은 자료흐름이 고려되도록 PROVIDED 절과 결합하여 CPN의 입력/조건 플레이스로 변환되고, FROM 절은 현재 상태 플레이스로, TO 절은 다음 상태 플레이스로 변환된다.

토큰은 초기상태 플레이스와 입력/조건 플레이스에 위치하게 된다. 토큰은 토큰의 개수를 나타내는 계수와, color로 구성된다. 초기상태 플레이스에는 임의의 color를 가진 하나의 토큰이 위치하게 되고, 각 입력/조건 플레이스에는 입력/조건에 따른 color를 가진 토큰이 위치하게 된다. I는 WHEN 절과 PROVIDED 절의 결합으로부터 변환된 입력/조건 플레이스와 FROM 절로부터 변환된 상태 플레이스가 함께 점화되어 트랜지션의 입력이 된다. 그리고 O는 TO 절로부터 변환된 다음 상태 플레이스인 D로 표현된다. 여기서 상태 플레이스인 C와 D는 Q라는 임의의 동일한 color를 갖고, 입력/조건 플레이스는 WHEN 절과 PROVIDED 절의 결합에 의해 color가 결정된다. PROVIDED 절을 입력/조건 플레이스로 표현함으로써 자료흐름을 부분적으로 하였다.

4. 사례 연구

이 논문에서는 Estelle로 기술된 TPO 프로토콜 명세를 CPN으로 변환하였다. 변환 수행 시 셀프루프는 실행가능 경로가 제거된 임의의 순서를 가지고 있다는 것과, 한 간선에 두 개 이상의 입력/조건이 존재하면 임의의 한 입력/조건을 사용한다는 가정을 하였다. 또한, 초기상태 플레이스의 indegree 예지는 라벨을 갖지 않는다. 이는 점화가능 트랜지션이 여러 개 존재함으로써 발생할 수 있는 OG의 상태폭발을 방지하기 위함이다.

이러한 조건을 만족하는 TPO의 CPN 표현은 그림 2에 나타나 있다.

그림 3은 그림 2를 Design/CPN으로 수행한 결과로 얻은 OG이다. OG에서 각 노드는 노드번호, 부모노드의 개수, 자노드의 개수에 대한 정보를 갖는다[1] 그림 4는 그림 3의 노드 1의 내부 정보를 보여주는 부분적인 descriptor이다. 그림 3의 모든 노드에 대해 그림 4와 같은 descriptor가 존재한다. 이 연구에서는 그림 3과 4에 [1]의 알고리즘을 적용하여, TPO의 UIO와 서브투어인 표 1과 표 2를 얻었다.

표 1은 Estelle로 표기된 프로토콜을 EFSM으로 변환한 방법에 의해 얻어진 UIO와 동일하다.

표 2는 CPN 표현 시 정해진 순서에 의해 셀프 루프를 고려하여 기존의 방법[1]과 동일한 서브투어가 생성되었다. TPO는 초기상태 플레이스인 idle에 셀프루프를 갖기 때문에 이 경로도 하나의 서브투어로서 표 2의 서브투어 1로 나타내었다.

이 연구에서 제안된 바와 같이 Estelle로 기술된 프로토콜 명세를 CPN 표현으로 변환하게 되면 UIO와 서브투어를 생성할 수 있는 정보를 자동적으로 얻을 수 있으며 부분적으로 자료흐름이 고려된 테스트 케이스를 얻을 수 있다.

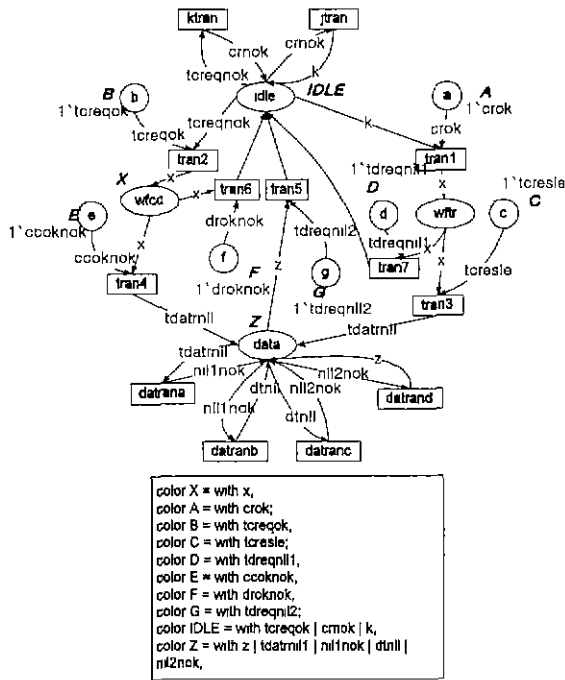


그림 2. Estelle로 표기된 TP0의 CPN 표현

5. 결론 및 향후 연구 방향

이 연구에서는 Estelle로 표기된 프로토콜 명세로부터 부분적으로 자료 흐름이 고려된 테스트 케이스를 얻기 위해 PROVIDED 절을 입력조건으로 추가하여 Design/CPN에 적용 가능한 CPN으로의 변환 방법을 제안하였다. 그리고 변환된 CPN을 Design/CPN에 입력하여 얻은 OG와 각 노드에 대한 내부 정보인 descriptor를 이용하여 UIO와 서브투어를 생성하였다. 또한 기존의 연구[1]에서 고려하지 않았던 셀프루프를 고려하여 Design/CPN에 적용하였다 기존의 테스트 케이스 생성 방법[3]은 Estelle로 기술된 프로토콜 명세를 EFSM으로 변환 후, UIO와 서브투어를 생성하는 각각의 알고리즘에 따라 테스트 케이스를 생성하였다. 이러한 방법은 여러 단계의 복잡한 과정을 수작업으로 수행하기 때문에 많은 오류의 가능성을 내포한다. 따라서 이 연구에서는 자동화된 틀을 이용하여 보다 간편하고 효율적으로 테스트 케이스와 추가 정보인 UIO를 얻을 수 있도록 CPN 변환 알고리즘을 제안하였다. 또한 PROVIDED 절을 입력/조건 플레이스에 나타냄으로써 부분으로 자료흐름을 고려하였다. 이러한 방법은 자동화된 프로토콜 테스트 환경의 기반을 제공한다.

향후 연구 방향은 프로토콜의 동작을 구체화 하는 다른 요소들을 추가 및 고려하여 자료흐름이 고려된 테스트 케이스를 추출하는 것과, 프로토콜을 직접 CPN으로 설계하여, 프로토콜의 설계 및 구현, 분석이 하나의 통합된 환경에서 자동적으로 수행될 수 있도록 하는 것이다.

참고문헌

[1] 이현정, 우성희, 이상호, "Design/CPN을 이용한 테스트 케이스 생성," 한국통신학회 하계학술발표 논문집 Vol. 17, No. 2, pp. 1285-1288, 1998

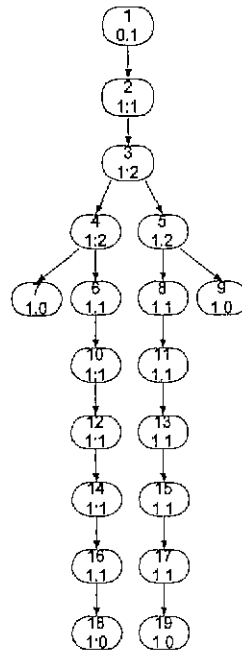


그림 3. TP0의 OG

1
New'idle 1: 1'tcreqnok
New'a 1: 1'crok
New'b 1: 1'tcreqok
New'wfcc : 1'empty
New'wfr : 1'empty
New'e 1: 1'ccoknok
New'c 1: 1'tcesle
New'data : empty
New'g 1: 1'tdreqnil2
New'd 1: 1'tdreqnil1
New'f 1: 1'droknok

그림 4. 그림 3의 노드 1에 대한 descriptor

표 1. 그림 3과 4에 의한 UIO

상태 플레이스	UIO
idle	tcreqnok
wfcc	ccoknok
wfr	tcesle
data	tdatrn1

표 2. 그림 3과 4에 의한 서브투어

서브투어	state
서브투어 1	idle - idle - idle
서브투어 2	idle - wfr - data - data - data - data - data - idle
서브투어 3	idle - wfr - idle
서브투어 4	idle - wfcc - data - data - data - data - data - idle
서브투어 5	idle - wfcc - idle

[2] K. Jensen, "Coloured Petri Nets," Vol. 1, 2nd Ed., Springer-Verlag, 1997.
 [3] 우성희, 오병호, 이상호, "Estelle로 표현된 프로토콜의 테스트 스위트 생성에 관한 연구," 정보과학회 춘계학술발표 논문집 21권 1호, pp. 463-466, 1994.
 [4] 오병호, 우성희, 이상호, "조건 문맥 변수를 고려한 제어 흐름도 생성 기법," 한국통신학회 논문지, 제 20권 12호, pp.3570-3579, 1995.
 [5] 이현정, 우성희, 오병호, 이상호, "실행불가능 경로가 제거된 테스트 케이스 생성," 정보과학회 춘계학술발표 논문집 25권 1호, pp. 443-445, 1998.