LTS로 명세화된 철도 신호제어용 프로토콜 검정 및 적합성시험
Protocol Verification and Conformance Test for Rail Signal Control Protocol specified in LTS

서미선* 김성운** 황종규*** 이재호***
Seo, Mi-Seon Kim, Sung-Un Hwang, Jong-Gyu Lee, Jae-Ho

ABSTRACT

As a very important part in development of the protocol, verification and conformance test for protocol specification are complementary techniques that are used to increase the level of confidence in the system functions as prescribed by their specifications. In this paper, we verify the safety and liveness properties of rail signal control protocol type 1 specified in LTS(Labeled Transition System) with model checking method, and experimentally prove that it is possible to check for the deadlock, livelock and rechability of the states and actions on LTS. We also propose a formal method on generation of conformance test cases using the concept of UIO sequences from verified protocol specification.

1. 서론

통신 소프트웨어에 대한 사용자의 요구사항이 복잡화, 다양화, 대형화되어짐에 따라 개발에 따르는 어려움은 더욱 증대되었고, 과거에 사용된 비정형적 방법(informal method)에 의한 개발은 많은 오류와 비효율성을 내포하고 있다. 따라서 개발에 소요되는 비용 및 시간을 절약하고 심각한 오류를 형식적 방법에 의해 검출하는 기술인 정형 기법(formal method)의 적용이 증가하는 추세이며, 특히 검정, 구현 및 시험 시에 전통적 자연어에 근거한 프로토콜 개발에 비하여 더 체계적인 방법을 제공한다[1].

사용자 요구사항과 명세와의 일치성을 구현 전에 확인하는 검정단계는 모든 프로토콜에 필수적인 정확성(correctness) 특성을 만족하는가에 대해 프로토콜 명세를 분석하는 과정으로서 모형검사(model checking)는 이러한 과정을 자동적, 형식적으로 검정하는 기술이다. 일반적으로 유한 상태 레이블 천이 시스템(LTS)으로 모델링된 concurrent 시스템 명세의 올바름을 검정하기 위한 모델 검정에 시제 논리에 기반을 둔 CTL(Computation Tree Logic)이 많이 사용되어 왔으나 해당 검정 알고리즘의 구현 및 적용 과정에서 시스템 내부 병렬 프로세스간의 상호작용으로 인해 시스템 요소의 증가에 따라 상태 수가 기하급수적으로 증가하는 상태폭발 문제가 제기되어 왔다. 그러나 modal mu-calculus 논리를 시스템의 안전성 및 필연성 특성 명세에 사용하면, 행위에 의한 순환적 정의가 가능하므로 상태 폭발 문제가 해결될 수 있다. 프로토콜 표준으로부터 관련 제품 구현에 있어 적합성시험은 구현 제품의 프로토콜 표준에 대한 적합성을 검정하는 시험으로, 프로토콜 표준으로부터 시험 계열을 생성하여 구현에 적용함으로써 적합성 여부를 판단하는 일련의 과정이다[2][3].

본 논문에서는 LTS(Labeled Transition System)로 명세화된 철도 신호제어용 프로토콜 모델의 안

* 부경대학교 정보통신공학과, 학생회원
** 부경대학교 정보통신공학과, 정회원
*** 철도연구원 선임연구원, 정회원

전성 및 필연성 특성을 모형검사 기법에 의해 검정하기 위해 대수적 명세기법인 modal mu-calculus를 사용하고 검정되어진 프로토콜 명세로부터 UIO(Unique Input/Output) sequence 기법에 의한 적합성 시험 계열 생성 방법을 제시한다.

2. 검정 및 시험대상

2.1 철도 신호제어 프로토콜 Type1

철도에 있어 신호제어시스템은 제한된 철도자원을 효율적이고 경제적으로 운용할 수 있도록 도와주며 열차의 안전 운행에 필수적인 시스템으로 철도 신호제어용 프로토콜에 의해 제어되어진다.

본 논문에서는 국내 철도 신호제어용 프로토콜 중의 하나인 중앙집중제어장치와 전자연동장치 간의 정보전송을 담당하는 역정보전송장치(LDTS : Local Data Transmission System)와 현장신호설비를 제어하고 감시하는 전자연동장치(EIS : Electronic Interlocking System) 사이의 정보전송방식(철도신호제어 프로토콜 Type1)을 검정 및 시험한다.

2.2 모델링 및 동작분석

LDTS는 EIS로 폴링 메시지와 진로설정이나 선로전환기 혹은 주신호기 등을 제어하는 제어 메시지를 전송하고, EIS는 LDTS로 현장 신호설비들의 상태정보 메시지와 제어 메시지에 대한 응답(ACK)을 전송한다. Fig.1은 Type 1의 흐름 제어를 프로토콜의 기능과 동작에 기반을 둔 I/O FSM(Input/Output Finite State Machine)으로 표현하고 각 상태의 설명을 나타낸 것이고, Fig.2는 정형명세를 위한 의미모델인 LTS로 모델링한 것이다.

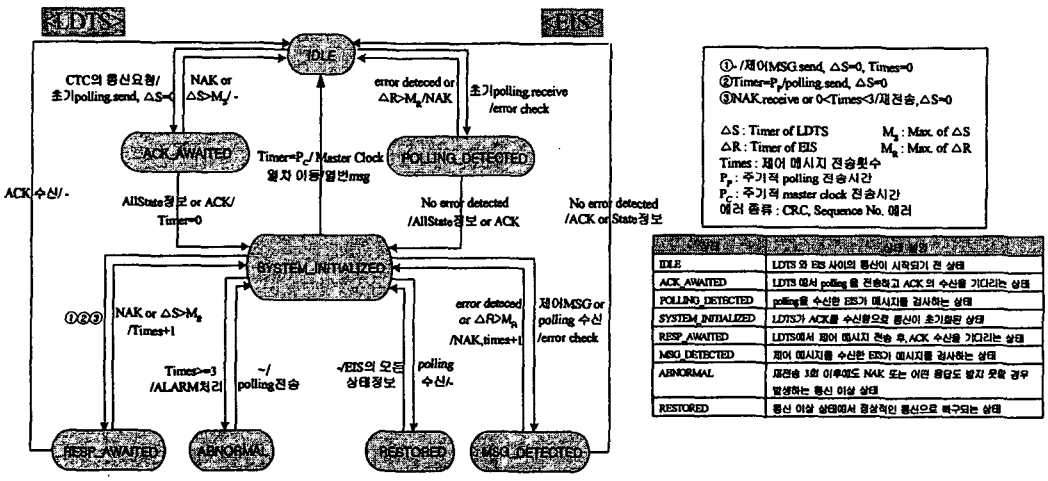

Fig.1 Type1의 I/O FSM 모델링 및 상태설명

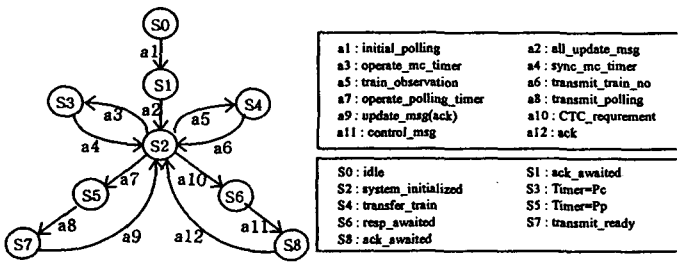

Fig.2 Type1의 LTS 모델링

3. Modal mu-calculus를 적용한 철도 프로토콜 검정

3.1 검정사항

통신 프로토콜이 적절한 기능을 하기 위해서는 프로토콜 각 해당 상태의 deadlock과 livelock 및 비정상적인 도달과 같은 잠재적 설계 에러가 없어야 하며, 사용자 요구사항과 일치하고 다른 프로세서와의 원활한 통신이 이루어져야 한다. Model checking에서 보다 구체적으로 검정해야 할 프로토콜의 특성은 다음과 같다.

- Deadlock : 한 상태에서 다른 어떤 상태로의 천이가 존재하지 않기 때문에 다음 행위를 할 수 없는 경우, 즉 그 상태에서 나가는 천이가 존재하지 않는다.
- Livelock : 프로토콜(명세) 상태들의 부분집합 내에서 그 상태들만을 무한히 반복적으로 천이하는 경우로서 그 부분집합 이외의 다른 상태로의 천이가 존재하지 않는다.
- Reachability : 초기상태로부터 프로토콜(명세)이 정의된 천이의 순서에 의해 정의된 다른 상태로 도달 가능하다면 그 천이와 상태에 대해 이 명세는 올바른 프로토콜(명세)이다.
- Liveness : 프로토콜의 어떤 정당한 특성(상태 또는 행위)이 결국 만족되어지는 것으로, 결국 도달되어야 하는 상태와 반드시 발생해야 하는 행위를 나타낸다.

3.2 Modal mu-calculus

시스템 특성 명세 언어인 modal mu-calculus는 시스템 상태의 부분 집합이 공통적으로 만족하는 논리식인 고정점을 가진 modal 논리이며, 연산자는 원자명제(atomic proposition), ∧(논리곱: conjunction), ∨(논리합: disjunction), [ ](필연성: necessity), < >(가능성: possibility), ν(최대고정점: greatest fixed point), μ(최소고정점: least fixed point)으로 구성되어 있으며, 일반화된 modal mu-calculus의 논리식은 다음과 같다.

$$\Phi ::= tt|ff|Z|\Phi 1 \wedge \Phi 2|\Phi 1 \vee \Phi 2|[k]\Phi|<k>\Phi|\nu Z.\Phi|\mu Z.\Phi \quad \cdots\cdots\cdots\cdots\cdots (1)$$

식(1)에서 tt는 모든 상태에 대해 참인 것을 나타내고 ff는 거짓임을 나타내며 Z는 명제적 변수, k는 천이 집합의 원소, Φ는 프로세스 특성을 나타낸 방정식이다. νZ는 시스템의 상태 집합들이 일반적으로 만족하는 특성인 최대고정점연산자이며, μZ는 시스템의 상태 집합들이 공통적으로 만족하는 최소고정점연산자이다[4].

3.2.1 안전성(Safety) 특성

프로토콜의 부당한 상태 즉, deadlock이나 livelock과 같은 상태를 배제하는 특성이다.

만약 정당한 상태를 나타내는 식이 Φ라면 상태에 대한 안전성은 νZ.Φ∧[-]Z로 표현되고 여기서 [-]는 모든 행위를 나타낸다. 위 식의 명제 값이 참이 된다는 것은 해당 LTS가 상태에 대한 안전성 특성을 가지고 있음을 의미한다. 또한 행위에 대한 안전성 표현은 νZ.[k]ff∧[-]Z이며, 식의 명제 값이 참이 된다는 것은 행위 k가 절대 발생하지 않음을 나타내며 LTS가 행위에 대한 안전성 특성을 가짐을 의미한다.

3.2.2 필연성(Liveness) 특성

Deadlock 및 livelock이 존재하지 않는다는 조건하에서 프로토콜이 초기 상태로부터 정의된 천이의 순서에 의해 결국에는 도달되어져야 하는 상태와 반드시 발생해야 하는 행위 즉, reachability와 liveness를 만족하는 특성이다.

LTS 모델로 표현된 프로세스의 상태에 대한 필연성은 μZ.Φ∨(<->tt∧[-]Z)로 표현되고, 이 식이 참이 된다는 것은 μZ.Φ와 (<->tt∧[-]Z)이 동시에 거짓이 될 수 없고 식에 해당되는 상태는 항상 도달되어져야 함을 나타낸다. 또한 LTS 모델로 표현된 프로세스에서 행위에 대한 필연성의 논리식은 μZ.Φ<->tt∧[k]Z로 표현하는데, 이 식이 참이 되기 위해 [k]Z는 반드시 참이 된다. 즉, 결국 행위 k는 발생할 수밖에 없음을 나타낸다. 필연성에 대한 식이 참이 된다는 것은 검정 대상 LTS가 필연성 특성을 가지고 있다는 것을 의미한다.

3.3 철도 프로토콜 검정 예

위에서 기술한 프로토콜의 안전성과 필연성 특성을 표현하는 modal mu-calculus 식으로부터 철도 신호제어용 프로토콜 type 1 LTS의 완전성을 검정하는 방법을 보인다. 예를 들어, 행위 a7이 발생하면 반드시 a8이 발생하고 deadlock과 livelock이 없음을 나타내는 modal mu-calculus의 논리식은 다음과 같으며, 식이 참이 되기 위해 μY. <-> tt와 [a8], [-]Z는 동시에 참이 되어야 한다.

$$\nu Z.[a7](\mu Y.<->tt \wedge [a8]Y) \wedge [-]Z \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (2)$$

위 식(2)에 model checking 알고리즘인 solve 알고리즘[5]을 적용하여 검정결과를 검출한다. Fig.3은 식(2)에서 최대고정점과 최소고정점을 사용하여 생성된 max block과 min block을 나타내고, Fig.4는 Fig.3에서 생성된 max block과 min block의 변수 Xi들의 천이 관계를 나타낸 edge-labeled directed graph G이다.

B1≡min{X1 = X2 ∧ X3	B2≡max{X5 = X6∧ X7
X2 = [a8]X1	X6 = [a7]X1
X3= <-> X4	X7 = [-]X5 }
X4 = tt }	

Fig.3 Max block, min block

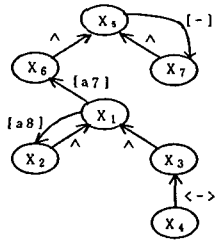

Fig.4 Edge-labeled directed graph G

Fig.5는 solve 알고리즘의 초기화 규칙에 의해 초기화된 bit-vector, counter 및 배열을 나타낸다. Fig.6은 배열 M[i]가 공집합(empty)이 될 때까지 갱신 알고리즘을 적용하여 bit-vector와 counter를 갱신한 결과로, 검정 대상인 deadlock, livelock 및 reachability를 판단할 수 있다.

X	X_1	X_2	X_3	X_4	X_5	X_6	X_7
S0	0	1	0	1	1	1	1
S1	0	1	0	1	1	1	1
S2	0	1	0	1	1	1	1
S3	0	1	0	1	1	1	1
S4	0	1	0	1	1	1	1
S5	0	0	0	1	1	1	1
S6	0	1	0	1	1	1	1
S7	0	1	0	1	1	1	1
S8	0	1	0	1	1	1	1

C	X_1	X_2
S0	1	0
S1	1	0
S2	1	0
S3	1	0
S4	1	0
S5	2	1
S6	1	0
S7	1	0
S8	1	0

M[1]=<<S0, X2>, <S1, X2>, <S2, X2>, <S3, X2>, <S4, X2>, <S6, X2>, <S7, X2>, <S8, X2>, <S0, X4>, <S1, X4>,<S2, X4><S3, X4>, <S4, X4>,<S5, X4>,<S6, X4>,<S7, X4>,<S8, X4>>
M[2]=< >

Fig.5 Bit-vector, counter와 배열 M[i]

X	X_1	X_2	X_3	X_4	X_5	X_6	X_7
S0	1	1	1	1	1	1	1
S1	1	1	1	1	1	1	1
S2	1	1	1	1	1	1	1
S3	1	1	1	1	1	1	1
S4	1	1	1	1	1	1	1
S5	1	1	1	1	1	1	1
S6	1	1	1	1	1	1	1
S7	1	1	1	1	1	1	1
S8	1	1	1	1	1	1	1

C	X_1	X_2
S0	0	0
S1	0	0
S2	0	0
S3	0	0
S4	0	0
S5	0	0
S6	0	0
S7	0	0
S8	0	0

M[1]=< >
M[2]=< >

Fig.6 결과 Bit-vector, counter와 배열 M[i]

Deadlock은 bit-vector의 요소에 의해 판단되는데, Fig.6의 결과에서 bit-vector의 요소는 모두 1로 갱신되어 deadlock이 발생되지 않았음을 알 수 있다. 또한 livelock은 관련된 상태의 counter 요소에 의해 판단되며 위의 결과에서 counter의 요소는 모두 0으로 갱신됨으로서 livelock이 검출되지 않았음을 알 수 있다. 즉 위 예시의 모든 요소가 max block과 min block을 만족하므로 Fig.2의 LTS 모델은 논리식 $\nu Z.[a7](\mu Y.<->tt \wedge [a8]Y) \wedge [-]Z$를 만족하는 완전한 프로토콜 모델임을 판단한다.

4. 적합성 시험

4.1 적합성 시험의 정의

일반적으로 적합성 시험은 주어진 명세 S를 기초로 하여 생성된 시험 계열(test cases)로서 구현 I(Implementation)가 원래 명세(혹은 표준) S(Specification)에 합당하게 구현되었는지를 시험하는 것으로 통신 프로토콜 제품 구현 과정에서 중요한 역할을 한다[6][7].

시험 계열 생성을 위한 일반적인 많은 방법들은 명세 S가 입출력 유한상태기계(I/O FSM) 형태로 표현된 프로토콜로부터 출발하는데, 프로토콜의 명세를 나타내는 I/O FSM 모델이 Fig.7과 같이 주어질 때 적합성 시험은 아래의 3단계로 이루어진다.

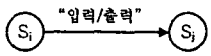

Fig.7 명세 I/O FSM

① 명세 I/O FSM의 스테이트 Si에 해당하는 스테이트로 구현 I/O FSM을 위치시킨다.

② 명세 I/O FSM에서 얻어진 시험 계열 "입력"을 구현 I/O FSM에 적용시킨 후 생성되는 "출력 "을 판단한다.

③ 구현 I/O FSM에서 생성된 출력이 명세 I/O FSM에서 기술된 "출력"과 같은지를 확인하고 도착한 스테이트가 명세 I/O FSM의 그것과 같은지를 검정한다.

4.2 UIO방법에 의한 시험 계열 생성

UIO(Unique Input/Output) 시퀀스는 시험하는 천이 후에 도착한 스테이트의 유일한 입력/출력 시퀀스를 시험 계열에 포함시켜 적용한 후, 구현 I/O FSM의 결과 스테이트를 확인하는 방법이다. I/O FSM으로 표현된 프로토콜 명세로부터 적합성 시험 계열 생성은 명세에 나타나 있는 각 천이에 도착 스테이트의 UIO 시퀀스를 concatenation하여 생성하는데 다음과 같은 식으로 나타낼 수 있다.

$$R_i \cdot \text{short-path}(S_0\text{-}S_i) \cdot T_{ij}@\text{UIO}(S_j) \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (3)$$

위 식에서 R_i는 시험대상을 초기화 상태로 보내는 심볼이며, short-path(S_0-S_i)는 초기상태 S_0에서 해당 시험천이에 대한 시작 상태까지 shortest path, T_{ij}는 시험되어야 할, 즉 명세 I/O FSM의 스테이트 S_i에서 S_j로 보내는 천이를 나타내고, UIO(S_j)는 도착 스테이트의 UIO 시퀀스를 나타낸다. 또 @는 concatenation 심볼이다. 여기서 UIO 시퀀스는 시험되는 천이에 의해 도착된 스테이트가 명세에서 원하는 올바른 S_j인가를 시험하는데 사용된다.

간결성을 위해 철도 프로토콜 type 1의 I/O FSM 모델 Fig.1의 각 천이를 표1과 같은 약어로 대체하고 UIO에 의한 시험 계열 생성방법을 적용하여 각 상태별 UIO 시퀀스를 구한 것은 표2와 같다. UIO 시퀀스를 사용하여 I/O FSM 모델로부터의 시험 계열 자동 생성을 위한 여러 가지 방법들이 제안되었다[8].

표2를 사용하여 식(3)을 적용하면 표3과 같은 철도 프로토콜 type 1에 대한 적합성 시험 계열이

표1 천이에 대한 약어표

해당천이	약어	해당천이	약어
CTC의 통신요청/초기polling.send, ΔS=0	A	NAK.receive or 0<Times<3/재전송, ΔS=0	K
NAK or ΔS>M_8/ -	B	NAK or ΔS>M_8/Times+1	L
error deteced or ΔR>M_8/NAK	C	Times>=3/ALARM처리	M
초기polling.receive/error check	D	-/polling전송	N
AllState정보 or ACK/Timer=0	E	-/EIS의 모든 상태정보	O
No error detected/AllState정보 or ACK	F	polling수신/-	P
Timer=P_C/Master Clock	G	error detected or ΔR>MR / NAK, Times+1	Q
열차 이동/train_num_msg	H	제어MSG or polling 수신/error check	R
-/제어MSG.send, ΔS=0, Times=0	I	ACK 수신/ -	S
Timer=P_N/polling.send, ΔS=0	J		

표2 Type 1 상태별 UIO 시퀀스

상태	UIO 시퀀스
IDLE(S0)	CTC의 통신요청/초기polling.send, ΔS=0 (A)
ACK_AWAITED(S1)	AllState정보 or ACK/Timer=0 (E)
POLLING_DETECTED(S2)	error detected or ΔR>M_8/NAK (C)
SYSTEM_INITIALIZED(S3)	-/제어MSG.send, ΔS=0, Times=0 (I)
RESP_AWAITED(S4)	ACK 수신/ - (S)
MSG_DETECTED(S5)	error detected or ΔR>M_8/NAK, Times+1 (Q)
ABNORMAL(S6)	-/polling전송 (N)
RESTORED(S7)	-/EIS의 모든 상태정보 (O)

생성된다. 표3의 ()내에 표현된 상태는 하나 이상의 천이에 대한 구별로서 출발상태와 도착상태를 나타낸다.

결과적으로 위에 나타낸 적합성 시험 계열의 입력 부분을 type 1에 대한 구현이 받아들이고 출력 부분을 생성하여 명세의 출력과 같은지를 판단하면 type 1의 구현이 type 1의 명세에 대해 정확하게 구현되었다는 적합성 시험결과를 확인할 수 있다.

표3 Type 1에 대한 적합성 시험 계열 생성

천이		천이에 대한 시험열	천이	천이에 대한 시험열
A		Ri·A·E	J	Ri·A·E·J·S
B		Ri·A·B·A	K	Ri·A·E·K·S
C		Ri·D·C·A	L	Ri·A·E·I·L·I
D		Ri·D·C	M	Ri·A·E·M·N
D		Ri·A·E·I	N	Ri·A·E·M·N·I
F	(S2→S3)	Ri·D·F·I	O	Ri·D·F·P·O·I
	(S5→S0)	Ri·D·F·R·F·A	P	Ri·D·F·P·O
G		Ri·A·E·G·A	Q	Ri·D·F·R·Q·I
H		Ri·A·E·H·A	R	Ri·D·F·R·Q
I		Ri·A·E·I·S	S	Ri·A·E·I·S·A

5. 결론 및 향후 연구 추진사항

본 논문에서는 LTS로 명세화된 철도 신호제어용 프로토콜 type 1 모델을 정형기법으로 명세화하고 안전성 및 필연성 특성을 대수적 명세 기법인 modal mu-calculus를 사용하여 검정하였으며, 검정되어진 프로토콜 명세로부터 UIO 방법에 의한 적합성 시험 계열 생성 방법을 제시하였다. 그 결과 해당 철도 프로토콜 type1은 안전성과 필연성을 모두 만족하는 규격임이 검정되었고 적합성 시험의 정확한 결과 또한 확인하였다.

본 논문에서 제시된 검정 방법과 시험 계열 생성 방법은 자연어에 근거한 수동적 검정과 시험 계열 생성에 비해 보다 신뢰성 있는 프로토콜을 개발하는데 사용되어질 것이고, 프로토콜 개발에 소요되는 비용과 시간을 최소화 시킬 것이다.

향후 연구 사항으로는 연구되어진 프로토콜 검정 및 적합성 시험 방법을 실제 프로토콜의 정확성을 분석하는데 사용할 수 있는 도구로 구현하는 과정에서, 프로토콜 검정기와 시험기의 검정결과를 GUI기능에 의해 Window상에 나타내도록 하는 것이다.

참고 문헌

1. D. Schwabe, Formal Techniques for the Specification and Verification of Protocol, Ph.D Thesis, Univ. of California Los Angeles, Apr., 1981.
2. ISO/IEC 9646-1, Information Technology - Open Systems Interconnection - Conformance Testing Methodology and Framework - Part1 : General Concepts, 1994.
3. R. Cleaveland, Tableau - Based Model-Checking in the Propositional Mu-Calculus, Acta Informatica 27 : 725-727, 1990.
4. Kenneth L. McMillan, Symbolic Model Checking, Kluwer Academic Publishers, Model Checking, 1996.
5. R. Cleaveland, B. Steffen, A Linear-Time Model-Checking Algorithm for the Alternation-Free Modal Mu-Calculus, Formal Methods in System Design 2(2) : pp.121-147, 1993.
6. Sung-Un Kim, INAP Protocol Conformance Testing, IEEE-AIN97, Colorado Springs, USA, May, 1997.
7. 김상기, 김성운, 정재윤, 형식기술기법에 의한 INAP프로토콜 적합성 시험계열 생성, 정보처리학회 논문집, 제4권 제1호, 1월, 1997.
8. Z. Kohavi, Switching and Finite Automata Theory, New York, McGraw-Hill, 1978.

후 기

본 연구는 철도청 철도기술연구개발사업으로 지원된 "신뢰성 확보를 위한 프로토콜 검정기 및 적합성시험 생성도구 연구" 과제의 연구결과의 일부입니다.