

## 컴파일러의 적합성 검증 추세

인소란\*

### I. 정의

컴파일러는 사용방법과 보유하고 있는 특성들이 동일하다 하더라도 구현환경에 따라서 각각 다르다. 즉 동일한 표준규격을 지닌 컴파일러라 할지라도 구현환경에 따라서 특수한 운영체제, 하드웨어, 그리고 구현전략등의 영향으로 서로 일치되지 못하고 있다. 따라서 컴파일러 검증(compiler validation)이란 대상 컴파일러가 그의 표준규격과의 적합한 정도(degree of conformance)를 결정하고자, 공식적으로 발표된 표준규격에 준하여 공식적으로 승인된 테스트와 테스트 방법을 통해서 실시되는 소프트웨어 테스트이다.

### II. 목적

구현시에 이용의 용이성, 개발 설계상의 특성, 구현 시스템(H/W, S/W)의 특성에 따라서 서로 다르게 구현된 컴파일러의 표준 규격의 범위에서 하나의 컴파일러가 취하는 행위를 결정하고자 하는데 있다.

### 목 차

- I. 정 의
- II. 목 적
- III. 출현배경
- IV. 필요성
- V. 현재의 추세
- VI. 검증종류
- VII. 검증서비스가 사용되는 대상
- VIII. 컴파일러 검증평가
- IX. 외국의 검증 실태
- X. 검증 시스템의 실례
- XI. 결 론

\* 정보전산연구실 선임연구원

### III. 출현배경

컴파일러 검증은 미국의 DoD의 요구에 따라 미국 해군성에 의해서 1968년 Cobol에 대한 Cobol Test suite개발과 더불어 미국에서 시작되었다. 테스트 슈트는 후에 ISO 1989-1978로서 ISO에서 승인한 ANSI X 3. 23-1974 표준과의 적합성(conformity)을 테스트하기 위해 개발되여졌고, 컴파일러들의 테스팅은 컴파일러 검증 서비스를 수행하는 목적을 가지고 특별히 설립된 정부 부서인 FSTC(Federal Software Testing Center)로 이동되었다.

연방재고조사 대상으로 들어오는 모든 Cobol 컴파일러들의 검증은 반드시 시행되어야한다(FIPS PUB21-1)는 내용에 대해 컴파일러가 따르는지 여부를 검증하는 공식적인 Audit routine set으로서 1977년 4월에 연방 증명서(Federal Register)의 공식 선언이 있었다. 이에 대한 상세한 내용은 1974년도판 CCVS(Cobol Compiler Validation System)의 version2.0 승인에 있다.

FCVS(Fortran Compiler Validation System)는 ISO 1539-1980(E)로서 수락된 X3.9-1978에 있는 기술규격에 기반을 둔 FSTC에 의해서 1978년에 발표되었고, 미국에서도 Fortran 컴파일러 검증을 FIPS PUB69의 지원으로 수행되었다. 1983년 FSTC에서는 프랑스의 BIADI, 독일의 GMD, 그리고 영국의 NCC로 하여금 동일한 테스트 슈트와 테스트 절차들을 사용하여 유럽에 있는 Cobol과 Fortran 컴파일러의 공식 검증을 수행하도록 허가하였다.

1985년 Cobol 표준안이 NCC에서 만들어졌을 때 BIADI와 GMD는 공동연구 과제로서 새로운 테스트 슈트의 규격을 인수받았고, NCC는 EC의 자금으로 실체적인 테스트 슈트 규격을 만들었다. Cobol85의 version1.0은 1986년 2

월 NCC에 의해서 발표되었고, 이는 FIPS PUB21-1을 대신하는 FIPS PUB21-2를 지원하는 공식적인 검증슈트에 의해서 수락되어졌다. 미국의 컴파일러 검증에 있는 서비스 본부의 구성은 1986년말 NBS의 ICST에 양도되었다.

### IV. 필요성

현재 사용되거나 개발되고 있는 컴파일러들은 그들이 따르고 있는 표준규격(; 국제적으로 승인된 표준규격, 또는 특별한 표준규격)에 따라서 구현되고 있다. 개발 대상인 컴파일러는 자신이 종속되어 있는 표준 규격중에서 mandatory한 요소는 모두 구현하여야 하며, 구현자의 구현시의 선택여부에 따라서 optional한 요소는 구현될 수도 있고 반대의 경우도 있다.

따라서 동일한 표준규격에 따라 구현된 동일 컴파일러 제품이라도 그들간의 호환성이 없게 되고, 이들을 이용하여 개발하는 응용 소프트웨어 제품들도 호환성뿐 아니라 여러가지 측면에서 더욱더 큰 side-effect효과를 지니게된다. 이로 인해 모두 완전한 컴파일러들이 그들의 표준규격에 따른다 해도 구현 제품들 간에 어떤 차이점들이 있는지 여부를 파악하여 제품들 간의 호환성을 유지할 필요가 있다. 예를 들어서 Ada컴파일러의 경우 Ada 표준 규격중에서 어떤 구현상의 종속부분들을 허용하는 요소들(; identifier의 최대 길이, 또는 정수형의 최대 값의 한계, 컴파일러들간에 다른 차이점들)은 특수한 운영체제, 하드웨어 선정등의 구현 전략의 특성에 따라서 선택된다.

따라서 제품화된, 또는 제품화 예정인 컴파일러는 공식적으로 발표되어서 다양한 응용 개발자들이 사용하기전에 반드시 적합성 검증을 하여 향후 발생되는 다양한 문제점들을 미연에

방지할 필요가 있다.

## V. 현재의 추세

최초로 컴파일러 검증은 미국에서 정부 측면의 구입자들이 요구함으로서 이를 지원하고자 수행되었다. 그러나 여러 유럽 국가에서도 컴파일러 구현자들에 의해 검증이 요구됨에 따라 표준 규격에 적합한 제품들의 이익을 중요시여기는 다른 주요한 구입자들 간에서도 이의 분위기가 확산되고 있다.

국제표준에 따라서 Cobol과 Fortran 컴파일러들을 검증하기 위해서 유럽을 통해 광범위하고 신뢰성이 있는 검증을 수행하는 검증 담당 센터간의 망구성이 추진되고 있고, 테스트의 질을 향상시키고 국제적으로 인가된 하나의 테스트 슈트와 방법을 지니게 되었다. 테스트 되여지기를 원하는 모든 제품들에게는 동일하고 명확한 검증절차들을 사용하는 검증 서비스의 신뢰성은 필수적이다.

영국의 NCC와 미국의 ICST간에는 이를 위한 협정이 수립되었고, 협정중에 있는 기술 문서에는 상세한 검증절차들을 이용하여 얻어진 테스트보고서와 검증의 상호 호환적인 인식을 받을 수 있는 공식적인 검증들을 구성함으로서 이를 유럽의 검증센터에서 사용할 수 있는 시스템을 제공해주고 있다.

〈표 5-1〉 외국에서의 컴파일러 검증의 현황

언어 국가	Cobol/Fortran	Pascal	C
미국		0	
독일	0	0	
영국	0	0	0
프랑스	0	0	0
이태리		0	
일본		0	
중국		0	

또한 영국의 NCC는 프랑스와 독일에서 Cobol과 Fortran 컴파일러 검증을 각각 운영할 수 있는 BIADI와 GMD를 인가하는데 지원하였다. 이러한 추세에 따라 〈표5-1〉에서와 같이 여러 국가에서 컴파일러 검증이 적극 추진되고 있다.

## VI. 검증 종류

일반적으로 H/W와 S/W제품에 대해 실시되고 있는 검증의 종류는 실시 목적에 따라서 Acceptance testing, Conformance testing, 그리고 Falsification testing으로 나누어 진다. 이들 각각의 특성은 다음과 같다.

### 1. Acceptance testing(인증성 검증)

인증성 검증은 소프트웨어 시스템이 자체 보유한 인증 평가 기준 및 항목에 만족되는지를 결정하고, 시스템을 승인하는지 사용자가 결정할 수 있도록 formal testing을 수행한다.

### 2. Conformance testing(적합성 검증)

적합성 검증은 표준 규격에서 요구한 특정한 요소들의 존재에 대해 대상 제품을 테스트한다. 검증 대상 제품이 적합한 구현 제품이라고 인증하는 범위까지 테스트를 실시하며, 테스트 슈트도 다음과 같이 5가지로 분류가 된다.

가. Passed test : 컴파일러가 예상했던 결과를 생성하는 테스트

나. Inapplicable test : 정상적인 상태에서 예상한 결과 보다는 다른 예외의 상태에서도 합법적으로 지원하는지 여부를 결정하는 언어의 특성을 사용하는 테스트

다. Failed test : 컴파일러가 컴파일러의 표

준 규격과의 비적합성이 있는 것을 증명하는 결과를 만들어 내는 테스트

라. Withdrawn test : 언어의 표준 규격과의 적합성을 조사하는데 사용되지 않고, 테스트가 부정확한 것을 찾아내는 테스트이다. 테스트는 어떤 타당하지 못한 테스트를 목적과 일치하는 것을 실패하거나, 언어의 비합리적 사용 또는 틀린 내용을 지니고 있으므로 부정확한 경우가 많다.

마. Split test : 컴파일러가 보유한 여러 회복 기능때문에 하나 또는 그 이상의 여러들이 어떤 테스트 부류에서 발견되지 못한 경우, 테스트는 발견되지 못한 여러들을 지닌 좀더 작은 테스트들의 집합체로서 다시 분리가 된다. 따라서 이들 분리된 split들은 각각 컴파일되어서 조사된 후, 모든 여러들이 컴파일러에 의해서 발견될 때까지, 또는 split당 어떤 여러가 있을 때 까지 splitting 처리과정을 계속하는 테스트이다. 즉, 테스트될 컴파일러의 크기때문에 번역과 수행을 할 수 없는 테스트 부류들을 처리하기 위해서 좀더 작은 subtest들의 집합으로 split한다. 예를 들어서 Ada의 경우 그에 속해 있는 테스트 부류 중 어떤 부류로 59개의 splits를 요구하는 경우도 있다.

### 3. Falsification testing(착오성 검증)

착오성 검증은 표준 규격의 많은 요구사항들의 테스트를 실시하고자 하는 테스트 케이스들을 사용하는 방법으로서, 테스트 슈트는 구현에 있는 여러들을 발견하는데 사용된다. 만일 여러가 발견되면 사용자는 표준 규격과 일치하지 않는 구현 내용을 정확하게 추론하게 된다. 그러나 반대로 여러가 없는 경우에도 이를 할 수 있다. 여러가 없다는 것은 구현이 표준

규격에 적합하든지, 또는 테스트 슈트가 여러를 발견하는데 충분하지 못하다고 파악되는 것을 뜻하고, 이로서 nonconformance를 증명할 수도 있다. (correctness tests의 증명들은 표준 규격과의 conformance를 증명하는데 사용한다.)

### 4. Field testing

표준 규격에 대하여 성공적인 적합성 검증을 수행하는데 필요한 절차를 향상시키고자 관련된 테스트 방법과 슈트를 조사하는 테스트이다.

### 5. Proficiency testing

interlaboratory test의 비교들을 통해서 Lab.의 testing 성능을 결정하는 테스트이다.

## VII. 검증 서비스가 사용되는 대상

- 제품 판매자가 그들 자신의 목적을 위해 검증해야 하는 컴파일러를 지닌 경우
- 판매자가 정부의 구매요구서(RFP : Request For Proposal)에 대응하여 검증되어진 컴파일러를 갖고자 하는 경우
- 구매 기관에 영향을 주는 정부기관
- 국가기관이 이미 사용중에 있는 컴파일러를 검증하기 원하는 경우
- 국가기관과 밀접한 관계를 지닌 다른 조직체에서 컴파일러의 검증을 사용하고자 하는 경우

## VIII. 컴파일러 검증평가

컴파일러의 검증 평가는 일반적으로 compilation listing과 결과보고서들을 분석함으로서 수행되어진다. 또한 컴파일러 여러는 문법에

러(syntax error)와 세멘틱 에러(semantic error)의 2가지 그룹으로 나누어지고, 문법 에러는 컴파일 시에 거의 발견되는 정적 에러(static error)이고, 세멘틱 에러는 수행시에 발견되는 동적 에러(dynamic error)이다.

## 1. 문법 에러

컴파일러에서 허용된 문법적(syntactical)이며 렉시칼(lexical)한 형태로 된 소스 코드 형태에서 발견되어진 불일치를 의미한다.

이는 컴파일러가 테스트될 언어 요소를 받아들이기 위해 컴파일러에 additional code를 요구하는 경우에의 하나의 에러가 되고, 특히 그러한 형태를 지닌 모든 변경요구들이 유지관리되도록 각각의 해당 컴파일러별로 수행 루틴을 지닌 변경 능력에 소스코드가 추가된다.

컴파일러는 동일한 토큰에 의해서 자신의 audit 루틴에 있는 어떤 소스코드를 거부하여서 이를 하나의 문법 에러로 취급하여 테스트는 그의 변경하는 능력을 사용하여 컴파일러의 audit루틴에서 제거된 에러를 발생시키는 언어 요소를 지니고 있다.

## 2. 세멘틱 에러

컴파일러의 audit 루틴들의 수행동안에 하나의 컴파일러 구현 제품에서 발견된 불일치이다. 이는 각각의 audit 루틴들을 수행하여 만든 테스트 결과(test results)에 실패들(failures)로서 나타나고, 이들을 위해 계산되고, 정확한 테스트 결과들을 테스트 결과 보고서에서 보여지게 한다.

# IX. 외국의 검증실태

다양한 컴파일러 개발의 선두자인 외국에서는 컴파일러의 적합성 검증 측면에서도 이미 5

~6년전 부터 환경 구축, 관련 기법 및 tool들을 개발완료하고 있다.

즉 Cobol/Fortran분야에서는 독일, 프랑스, 영국등에서 이미 완료하였고, Pascal은 영국, 프랑스, 독일, 이태리, 미국, 일본, 중국등 여러 나라에서 취급하고 있다. 또한 C 분야에 대해서도 프랑스, 영국, 이태리등에서 취급하고 있다(표 9-1 참조).

〈표 9-1〉 외국의 검증기관 현황

언어 국가명	Cobol/Fortran	Pascal	C
독 일	GMD	GMD	
프 랑 스	BIADI	AFNOR	AFNOR
영 국	NCC	BSI	BSI
이 태 리	IMQ	IMQ	
미 국		NBS	
일 본		JMI	
중 국		CSBS	

- \* GMD : Gessellschaft für Mathematik und Datenverarbeitung mbH Z2.2
- \* BIADI : Bureau Inter Administration de Documentation Informatique
- \* AFNOR : Association Francais de Normalisation
- \* NCC : The National Computing Centre Ltd.
- \* BSI : British Standards Institution
- \* IMQ : Istituto Italiano del Marchio di Qualita
- \* NBS : National Bureau of Standards
- \* JMI : Japan Machinery and Metals Inspection Institute
- \* CSBS : State Bureau of the People's Republic of China

# X. 검증 시스템의 실례

본고에서는 미국 AT&T에서 설계 구현시킨 검증 시스템을 실례로 하였다. 메인 시스템은 AT&T 3B2/400이고, 운영체제는 Unix이며, 검증대상 컴파일러는 Ada Release 1.0이다.

## 1. 특성

테스트는 구현과 종속되어 있으나, Ada 표준

에 의해 허용된 행위들을 나타내 주는 6가지의 테스트 부류들(test classes)이 사용되었다.

## 2. 검증 목적

컴파일러가 합법인 언어 구조를 적합하게 구현한 것과, 컴파일러가 비합법적인 언어구조를 나타냄으로서 이를 거부하는지를 테스트함으로서 Ada 표준에 대해 컴파일러의 적합성을 보장하고자 하는데 있다.

- Ada 표준과 일치하지 않는 컴파일러에 대해서 지원된 어떤 언어 구성을 나타내고자
- Ada 표준에 대해서 요구되었으나 지원되지 않는 언어 구성을 나타내고자
- 구현에 종속된 행위가 Ada 표준에 의해 허용된 것임을 결정하고자

## 3. 검증 절차

검증절차는 Ada 컴파일러를 입력으로 하여 ACVC(Ada Compile Validation Capability), 표준화된 테스트 슈트를 제출한 후, 그의 결과들을 대상으로 평가하는 것을 포함하고 있다.

## 4. 검증 부류

ACVC는 합법적인 Ada 프로그램과 비합법적으로 Ada 프로그램등 2개에 대해 6가지 테스트 부류(A, B, C, D, E, L)를 지니고 있다.

- 테스트 부류 A, C, D, E는 executable하고, 특별한 프로그램 단위는 수행하는 동안에 그들의 결과를 기록하는데 사용되여진다.
- 테스트 부류 B는 compilation error들을 만드는데 사용되고,
- 테스트 부류 L은 link error들을 만드는데 사용된다.

## 5. 조직 구성

AVO(Ada Validation Organization)에 의해 수립된 정책들과 절차들에 따라서 AVF(Ada Validation Facility)의 지도하에서 구성되었고, on-site 테스팅은 AT&T까지 구성되어 수행되었다.

## 6. 검증 시스템 구성

- 대상제품 : Unix Ada release 1.0
- ACVC version : Version 8
- Host computer 및 target computer :
  - 컴퓨터 시스템기종 : AT&T 3B2/400 (with MAU)
  - 운영체제 : Unix system V release 3.1
  - 주기억장치 용량 : 4MB

## 7. 구현 측면에서의 특성

테스트 부류 D와 E는 컴파일러 구현 측면에서의 차이점을 조사하는데 사용되나, 그외의 다른 테스트 부류들에 있는 테스트들은 자체의 하나의 구현제품을 특징화하는데 사용되고 있다.

Ada 표준 규격에 비해서 대상제품은 〈표 10-1〉과 같은 특성을 지니고 있다.

〈표 10-1〉 테스트 부류별 테스트 결과

테스트결과	테 스 트 부 류						합계
	A	B	C	D	E	L	
Passed	69	860	1164	17	11	33	2154
Failed	0	0	0	0	0	0	0
Inapplicable	0	27	204	0	2	13	226
Withdrawn	0	7	12	0	0	0	19
합 계	69	874	1380	17	13	46	2399

〈표 10-2〉 검증시스템의 구성도

검증시스템	적합성 검증을 수행하기 위한 수행절차와 관리에 대한 자체의 법규를 지닌 시스템
검증 몸체	검증 시스템의 운영을 하거나 accredit하고자, 필요한 능력과 신뢰성을 지닌 정부적 또는 비 정부적 공평한 몸체이며, 시스템의 기능에 관련된 모든 parties의 관심들이 표현되어 있는 부분이다.
검증 표시	검증되어진 제품 또는 서비스를 identify하는 검증 몸체의 validating sign, symbol, letter가 있다.

## XI. 결 론

현재 국내외적으로 각종 다양한 정보 및 소프트웨어 tools의 설계, 개발 및 구현되고 있다. 이들 중에서 하나의 분야인 컴파일러도 역시 외국의 선두 주자인 만큼 다양한 성능을 지닌 제품들이 발표되었다. 특히 어떤 컴파일러를 상용화시키고자 하는 경우 그의 표준 규격은 항상 보유하고 있는데 실제 설계자가 구상하여 작성된 표준 규격이 상용화에 참여된 개발자들 간의 규격과 적합성이 일치되지 못하면 이 제품은 융통성과 확장성, 그리고 이용의 용이성 측면에서 상당한 overhead와 난점이 있다고 본다. 그래서 외국에서는 이미 5~6년 전부터 적합성 검증의 필요성을 인지하고 관련된 소프트웨어와 장비 등 시험 환경을 구축하고 있다. 즉 미국과 프랑스, 영국, 이태리 등 유럽과 중국과 일본 등 아시아 지역에서 이를 적극 추진하고 있다. 이에 부응하여 현재 우리나라에서도 정보 및 각종 소프트웨어 제품이 외국 수준에 대응될 수 있는 각종 다양한 제품이 출현되는 상황에서 적합성 검증에 대한 준비를 하는 측면에서 외국과 동일하게 적합성 검증의 구축이

요구되고 있다.

따라서 본고에서는 컴파일러의 적합성 검증에 관련된 사항을 외국의 추세를 토대로 기술적인 측면에서 기술 동향분석을 하는데 중점을 두어서 작성되었다.

### \* 용어 해설

- certification—제품이나 서비스가 표준 규격에 적합한지에 대해 대해 테스트를 통해 실시하여 검증하는 절차
- validation—특수한 표준 규격이나 집합을 지원하고자 만들어진 프로그램이나 소프트웨어의 정확도(correctness)를 결정하는 절차
- verification—일치성(consistency), 완전성(completeness), 정확성(correctness)를 결정하는 절차
- error—테스트하고자 하는 자신의 규격과 일치되는 검증 제품(IUT)에 대한 실패(failure)
- nonconformance—표준 규격에 대한 적합성을 위해 보장하고자, 테스트들의 계획된 수량을 통과시키지 못한 상태에서 undergone conformance testing을 지닌 제품
- conforming implementation—conformance 요구사항들을 만족하는 IUT가 테스트 슈트의 pass/fail 문서에서 진술한 capabilities 및 options에 일치함
- test method—제품의 하나 또는 그 이상의 정해진 특성들을 결정하기 위해서 지정된 기술적인 절차
  - 테스트 소프트웨어가 포함되어 있다.
  - 기술적인 절차의 부분은 어떤 프로그래밍 언어에서 있는 문장들에 의해서 기술되어 있다.

- test report—테스트 결과들과 테스트에 적합한 다른 측면의 정보를 나타내는 문서
- test suite—IUT에 대한 적합성 테스트는 수행에 필요한 테스트들의 완전한 집합으로서, 이들이 수행되어질 수 있는 형태로 되어 있을 때 수행되어질 순서를 결정하는 정보가 있다.
- summary report—테스트 후에 IUT에 대한 상세한 내용 및 테스트 되어진 환경들을 기록한 내용을 중심으로 작성된 문서이다. 수행된 테스트 케이스, 테스트 목적, 테스트 분석, 입증들에 대한 전반적인 요약서이다.
- IUT—테스트 중에 검토되어질 예정인 표준의 부분으로서, 하나의 또는 그 이상의 특성들로 구현된 제품
- SUT—구현 제품이 들어 있는 컴퓨터의 하드웨어와 소프트웨어
- verdict—구현 제품에 대한 적합성을 위한 pass문, fail문, inclusive문으로서 수행되어진 테스트 케이스를 만든다.
- fail verdict—테스트 규격에 “fail”로서

명명되었거나, 분류가 되어진 결과가 관측된 결과와 일치시 verdict가 주어진다.

- pass verdict—테스트 규격에 “pass”로서 명명된 결과와 관측된 결과가 일치시에 verdict가 주어진다.
- inclusive verdict—관측된 결과가 테스트 케이스 규격에 inclusive로서 분류되었거나, 명명되어진 결과와 일치하든지, 또는 테스트 케이스 규격에 어떤 명해진 결과와 일치하지 않을 때 verdict가 주어진다.

## 참 고 문 헌

1. *ada compiler validation summary report*, AT & T
2. *Ada compiler validation capability implementor's guide*, Softech Inc., Dec. 1984
3. *Conformance testing service*, Commission of the European Communities—DG XIII—Telecommunications, Information Industries and Innovation