TECHNICAL REPORT 9547

Published 1988-04-01

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION • МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ • ORGANISATION INTERNATIONALE DE NORMALISATION

Programming language processors — Test methods — Guidelines for their development and acceptability

Processeurs de langage de programmation — Méthodes d'essai — Lignes directrices pour leur élaboration et acceptabilité

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work.

The main task of ISO technical committees is to prepare International Standards. In exceptional circumstances a technical committee may propose the publication of a technical report of one of the following types:

— type 1, when the necessary support within the technical committee cannot be obtained for the publication of an International Standard, despite repeated efforts;

— type 2, when the subject is still under technical development requiring wider exposure;

— type 3, when a technical committee has collected data of a different kind from that which is normally published as an International Standard ("state of the art", for example).

Technical reports are accepted for publication directly by ISO Council. Technical reports types 1 and 2 are subject to review within three years of publication, to decide if they can be transformed into International Standards. Technical reports type 3 do not necessarily have to be reviewed until the data they provide is considered no longer valid or useful.

ISO/TR 9547 was prepared by Technical Committee ISO/TC 97, *Information processing systems*.

This type 3 Technical Report has been prepared to provide guidelines for those developing test methods for programming languages and also to aid those developing the programming languages themselves. It should facilitate the preparation of specific test methods which will help implementors to produce standard-conforming language processors, thereby benefitting end users.

UDC 681.3.068 : 620.1 **Ref. No. ISO/TR 9547 : 1988 (E)**

Descriptors: data processing, computer programs, programming languages, language processors, tests.

Printed in Switzerland

Price based on 6 pages

CONTENTS

0. INTRODUCTION

The programming language area is a living topic: new languages are developed, old ones are revised. In those two cases, conformity requirements should not be forgotten; therefore, test considerations should be included in the development of language standards and their revisions.

The aim of this document is to provide guidelines for the development of a test method based on a set of test programs and precise steps that should be taken in order for a test method to get formal approval from ISO. These guidelines should be read in conjunction with a given language standard so as to produce a specific test method for programming language processors.

These guidelines are written for test method developers and for those who will approve these test methods in ISO. The availability of specific test methods will help implementors in producing standard conforming language processors, thereby benefiting end users.

When reading these guidelines, it should be borne in mind that programming language standards have not yet reached a level of precision and completeness to allow conformity tests to be produced for every aspect and feature of a language described in a progamming language standard. Testing issues are rarely the primary objective of standardization committees, and this may cause problems in applying these guidelines strictly to any given programming language standard.

1. SCOPE AND FIELD OF APPLICATION

These guidelines describe a methodology for determining whether a programming language processor possesses the required characteristics stated in the International Standard for the particular programming language for which it is intended.

Assessment of conformity of a language processor can be carried out with the "Test Suite Methodology". Other methodologies are not excluded but are not described here.

2. DEFINITIONS

For the purpose of these guidelines, the following terms apply.

2.1 CONFIGURATION : host and target computers, any operating system(s) and software used to operate a processor.

2.2 EXTENSION : a facility in the implemented language that is not given in the language standard but that does not cause any ambiguity or contradiction when added to the language standard (although, in some languages, it may serve to lift a restriction).

2.3 IMPLEMENTATION DEFINED : dependent on the processor but required by the language standard to be defined and documented by the implementor.

2.4 PROCESSOR : a compiler, translator or interpreter working in combination with a configuration.

2.5 TEST : technical operation that consists of the determination of one or more characteristics of a given product, process or service according to a specified procedure. (ISO/IEC Guide 2-1986)

2.6 TEST METHOD : specified technical procedure for performing a test. (ISO/IEC Guide 2-1986)

2.7 TEST PROGRAM : a sequence of characters intended to be submitted to a processor in order to determine whether or not this processor exhibits a specific instance of a certain property.

2.8 TEST REPORT : document that presents test results and other information relevant to a test. (ISO/IEC Guide 2-1986)

2.9 TEST SUITE : a reference set of test programs that is designed to assess conformity of a processor with a language standard.

2.10 TEST TOOLS : any additional means that can improve the efficiency, the reliability and the ease of use of the different phases of testing (e.g. implementation of the test suite, ensuring integrity, processing of the test suite, collecting test results, analysis of test results, producing a test report).

2.11 REQUIRED DOCUMENTS : the set of documents required by the programming language standard.

2.12 SUBSET : a subset S of programming language L is a programming language such that every program in S
- is also a program in L and
- has the same meaning in S as it has in L.

3. OBJECT TO BE TESTED

As a processor only works in combination with a configuration, some conditions should be taken into account.

3.1 The processor to be tested, including selected options, and the configuration relevant to the testing should be fully specified.

3.2 A single copy of a processor should be tested on a specific configuration.

4. DESCRIPTION OF TEST METHODS

In this context the technical procedure utilizes a test suite, test tools and possibly data.

4.1 TEST SUITE

A test suite should be designed to test conformity of processors by submitting test programs to them. Each test exercises some rules of the language standard and their interaction. A test suite should be designed for conformity testing and not for assessing other aspects of a processor.

The test suite should, wherever feasible, cover all aspects of the language standard and investigate implementation issues as far as conformity is concerned.

The test suite should not be too large: the economical aspects should be taken into account (e.g., relative costs of conformity testing versus development cost of an implementation).

The test suite should be written in such a way that it can be readily maintained under version control and subject to a review procedure.

The test suite should be designed so as to take into account the possible levels or options specified in the language standard. Thus the test suite should be modular.

The test programs should, as far as possible, be independent of each other, and their sequence of execution should not influence each other's results. Where this is not possible or desirable, due to some aspects of language design, the relationship between any tests (or source code modules) which depend upon each other should be clearly documented.

Each test program should have a single objective related to the standard's requirements. When this is not practical results of individual objectives must be readily identifiable. Test programs should produce a result in accordance with the stated objectives.

The test suite should contain test programs that are in accordance with the rules of the language standard. It should also contain test programs that are not in accordance with the rules of the language standard, in those situations where the language standard specifies syntactic or semantic properties that must be rejected. Test programs should be made as far as possible self-checking and hence report the success or failure by a message (which should aid automated production of a report).

The test suite should allow for the use of parameterized values that take into account the configuration used with the processor.

4.2 TEST TOOLS

Provision should be made for the development of test tools. Automation of both the testing and the analysis of results is essential to facilitate rigorous testing on an economical basis.

4.3 TECHNICAL PROCEDURES

4.3.1 OVERALL SCHEME FOR THE TESTING

The test method should define the overall scheme for the testing as described below.

Using the processor to be tested, a test suite should be run on a designated configuration. The results (pass or fail) of testing a specific processor on a specific configuration (all components being completely defined) will be reproducible if no modification has occurred to any component of the configuration.

The information given in the required documents should be inspected for conformity to the language standard and should be compared with the results of testing.

The output from a processor that has been tested with the test suite should be analysed in accordance with clearly defined rules. These rules should give criteria for objective evaluation of all possible outputs, which may include no output at all, for each test program in the test suite.

The results of testing conformity to the standard need to be analyzed and compiled in a report. The report should provide and summarize all the information pertaining to the testing of conformity to the language standard (set up of testing, actual testing, major events during testing). The test report should also detail language features not testable, with reason where appropriate, for the implementation under test. The analysis and report production should be as automated as possible. The degree of automation should depend upon the kind of testing and the format of results.

4.3.2 RELATED DOCUMENTATION

The test method should specify documentation including:

- the description of test classification together with the objectives of each class;
- the order of the execution of the tests when required;
- a list of language features not covered by the test suite.

5. GUIDELINES FOR THE DEVELOPMENT OF THE TEST METHODS

Several points should be taken into account when preparing a test method. Those points are listed below.

5.1 CONSIDERATION OF EXISTING TEST METHODS

Test methods that have been used for other processors should be examined to determine how successfully they have achieved the design objectives for which they were constructed. When the design objectives for a new test method are similar to those for existing test methods, consideration should be given to the feasibility of adapting the applicable construction principles to the new test method.

5.2 CRITERIA FOR DESIGN OF TEST SUITES

Each test program should be written in such a way that its understanding is facilitated by the use of clear documentation in it that includes:

- comments that are clear, concise and not redundant from one test program to another;

- reference to clauses of the language standard;

- clear statements of assumptions made in test suite design.

The test suite should be written in such a way that its understanding is facilitated by the use of clear coding style such as the use of:

- naming conventions for test programs;

- mnemonic identifiers;

- consistency of coding style throughout the test suite;

- upper and lower cases.

The test suite should be written in such a way that it takes into account technical aspects specified in the language standard that are implementation defined, for example :

- implementation parameters (for example, use of the largest integer supported);

- maximum and minimum parameter values with a reasonable selection in between (for instance, measure of depth nesting);

- use of features that do not exist in subsets;

- floating-point precision;

- use of files and external data;

- input and output facilities.

The test suite should be written in such a way as to minimize the use of features which could be restricted by configuration or implementation characteristics. Where such features are used in the test suite for assessing other aspects of conformity, reasonableness should be exercised with regards to dependency limitations (e.g., numerical values, size of arrays).

The test suite should be also designed to use a minimum set of simple statements of the language to test complex features. Complex language constructs should only be used where specifically needed in a particular testing purpose. Programs that do not conform to the language standard should deviate from it only as necessary in order to satisfy a specific testing objective.

5.3 TESTABILITY

The test method should be designed in such a way that it should not be too difficult to implement. The test method should be a useful tool for implementors, hence incorporating the use of a test suite for testing a compiler during development and maintenance should not be too expensive.

The volume of output created during the application of the test method should be kept to a minimum.

5.4 LANGUAGE FEATURES NOT COVERED BY THE TEST SUITE

Some features of the language may not be testable.

Other features of a language standard may have been excluded from the scope of the test method, for practical or other reasons (e.g., avoidance of undue size or complexity of the test suite).

Where features of the language are not covered by the test suite, the test method should include a statement (or list) to this effect.

Particular problems may be experienced by test suite implementors with respect to language extensions. In general, where the language standard permits extensions, but does not constrain them, testing of extensions should be considered outside the scope of the test method. Where extensions are not permitted, it may be particularly difficult to test thoroughly for the absence of extensions. Special care should be taken to document the extension testing strategy (if any) in the test method.

5.5 SUBSET TESTING

The test method should be designed in such a way that subsets (when allowed by the language standard) can be tested by running either the same set or a subset of the test programs.

5.6 PORTABILITY REQUIREMENTS

The test suite should be portable to different configurations. A small configuration should not be penalized. Thus, each single program should be able to run on a small configuration, as well as on a large one. The result of testing (i.e., pass or fail) should not be influenced by the program's size.

5.7 TEST TOOLS

Automated test tools should, if possible, be written using the language whose processor is being tested, and should be portable to different configurations.

6. MAINTENANCE AND REVISION OF TEST SUITES

Provisions should be made for maintenance and revision of test suites through a public review and according to a published timetable.

7. AVAILABILITY

Test suite and tools should be available on one or more media formats that are generally used within the industry. Consideration should be given to the use of ISO standards.

8. ACCEPTABILITY OF TEST METHODS

The following points should be considered for a specific test method to achieve acceptability.

8.1 CRITERIA FOR ACCEPTABILITY OF THE TEST METHOD

Several criteria exist.

1) The related language should be an ISO standard.

2) The test objective and test programs should relate to conformity requirements of the standard.

3) The test suite should be complete in the sense that it tests all the points of the language, wherever possible.