

# INTERNATIONAL STANDARD

**ISO  
8807**

First edition  
1989-02-15

---

---

## **Information processing systems — Open Systems Interconnection — LOTOS — A formal description technique based on the temporal ordering of observational behaviour**

*Systèmes de traitement de l'information — Interconnexion de systèmes ouverts —  
LOTOS — Technique de description formelle basée sur l'organisation temporelle de  
comportement observationnel*



Reference number  
ISO 8807 : 1989 (E)

## Foreword

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. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

Draft International Standards adopted by the technical committees are circulated to the member bodies for approval before their acceptance as International Standards by the ISO Council. They are approved in accordance with ISO procedures requiring at least 75 % approval by the member bodies voting.

International Standard ISO 8807 was prepared by Technical Committee ISO/TC 97, *Information processing systems*.

Users should note that all International Standards undergo revision from time to time and that any reference made herein to any other International Standard implies its latest edition, unless otherwise stated.

Annex A forms an integral part of this International Standard. Annexes B, C, D and E are for information only.

## CONTENTS

	Page
0 Introduction . . . . .	1
0.1 General . . . . .	1
0.2 FDTs . . . . .	1
0.3 The requirement for standard FDTs . . . . .	1
0.4 The objectives to be satisfied by an FDT . . . . .	1
0.5 The origin of LOTOS . . . . .	2
0.6 The structure of this International Standard . . . . .	2
1 Scope and field of application . . . . .	3
2 References . . . . .	3
3 Conformance . . . . .	3
4 Basic mathematical concepts and notation . . . . .	5
4.1 General . . . . .	5
4.2 Sets . . . . .	5
4.3 Lists . . . . .	6
4.4 Strings . . . . .	6
4.5 Relations and functions . . . . .	6
4.6 Backus-Naur Form . . . . .	7
4.7 Syntax-directed definitions . . . . .	8
4.8 Derivation systems . . . . .	10
5 Model . . . . .	13
5.1 Introduction . . . . .	13
5.2 Many-sorted algebras . . . . .	13
5.3 Labelled transition systems . . . . .	13
5.4 Structured labelled transition systems . . . . .	14
6 Formal Syntax . . . . .	15
6.1 Lexical tokens . . . . .	15
6.1.1 General . . . . .	15
6.1.2 Basic characters . . . . .	15
6.1.3 Reserved symbols . . . . .	15
6.1.3.1 Word symbols . . . . .	15
6.1.3.2 Special symbols . . . . .	17
6.1.4 Identifiers . . . . .	17
6.1.5 Requirement . . . . .	18
6.1.6 Comments . . . . .	18
6.1.7 Token separators . . . . .	18
6.1.8 Requirement . . . . .	18
6.2 Specification text . . . . .	18
6.2.1 specification . . . . .	18
6.2.2 definition-block . . . . .	18
6.2.3 data-type-definitions . . . . .	19
6.2.4 p-expressions . . . . .	19
6.2.5 sorts, operations and equations . . . . .	19
6.2.6 process-definitions . . . . .	20
6.2.7 behaviour-expressions . . . . .	20
6.2.7.1 general structure . . . . .	21
6.2.7.2 local-definition-expressions . . . . .	21
6.2.7.3 sum-expressions . . . . .	21

6.2.7.4	par-expressions . . . . .	.21
6.2.7.5	hiding-expressions . . . . .	21
6.2.7.6	enable-expressions . . . . .	.21
6.2.7.7	disable-expressions . . . . .	.22
6.2.7.8	parallel-expressions . . . . .	22
6.2.7.9	choice-expressions . . . . .	22
6.2.7.10	guarded-expressions . . . . .	.22
6.2.7.11	action-prefix-expressions . . . . .	22
6.2.7.12	atomic-expressions . . . . .	22
6.2.8	value-expressions . . . . .	23
6.2.9	declarations . . . . .	.23
6.2.10	special-identifiers . . . . .	23
7	Semantics . . . . .	.25
7.1	Introduction . . . . .	.25
7.1.1	Structure of the static semantics definition . . . . .	25
7.1.2	Structure of the dynamic semantics definition . . . . .	25
7.1.3	Structure of clause 7 . . . . .	.26
7.2	General structures and definitions . . . . .	26
7.2.1	Names and related functions . . . . .	26
7.2.2	Algebraic specifications, terms, and equations . . . . .	27
7.2.2.1	Signature . . . . .	.27
7.2.2.2	Terms . . . . .	.27
7.2.2.3	Equations . . . . .	.27
7.2.2.4	Conditional equations . . . . .	28
7.2.2.5	Algebraic specifications . . . . .	28
7.2.3	Canonical specifications . . . . .	.28
7.2.3.1	Introduction . . . . .	.28
7.2.3.2	Behaviour-expression-structure . . . . .	28
7.2.3.3	Behaviour specification . . . . .	29
7.2.3.4	Canonical LOTOS specification . . . . .	.29
7.3	Static semantics . . . . .	.29
7.3.1	Introduction . . . . .	.29
7.3.2	General structures and definitions . . . . .	30
7.3.2.1	Scope . . . . .	30
7.3.2.2	Extended identifiers . . . . .	.30
7.3.2.3	The interpretation of extended identifiers . . . . .	31
7.3.2.4	Functionality . . . . .	.32
7.3.2.5	Data-presentation . . . . .	.32
7.3.2.6	Parameterized data-presentation . . . . .	32
7.3.2.7	Non-overlapping data-presentation . . . . .	33
7.3.2.8	Signature morphism . . . . .	33
7.3.2.9	Data-presentation morphism . . . . .	33
7.3.2.10	Environments . . . . .	34
7.3.2.11	Standard library . . . . .	34
7.3.2.12	Complete data-presentation . . . . .	35
7.3.2.13	Valid dependence order . . . . .	35
7.3.3	Reconstruction of terms . . . . .	35
7.3.3.1	Value-atoms . . . . .	.36
7.3.3.2	Value-atom, position, and argument-list of a value-expression . . . . .	.36
7.3.3.3	Operation-assignment . . . . .	37
7.3.3.4	Consistent operation-assignment . . . . .	37
7.3.3.5	Explicit sort indication . . . . .	.37
7.3.3.6	Sound operation-assignment . . . . .	37
7.3.3.7	Generated operation-assignments . . . . .	37

7.3.3.8	Scopes of an operation-assignment . . . . .	38
7.3.3.9	Minimal operation-assignment . . . . .	38
7.3.3.10	Reconstruction of a term . . . . .	38
7.3.4	Flattening of a LOTOS specification . . . . .	39
7.3.4.1	Introduction . . . . .	39
7.3.4.2	Flattening of specification. . . . .	39
7.3.4.3	Flattening of data-type-definitions . . . . .	40
7.3.4.4	Flattening of process-definitions . . . . .	48
7.3.4.5	Flattening of behaviour-expressions . . . . .	49
7.3.4.6	Flattening of identifiers . . . . .	55
7.3.5	Functional structure of the flattening function . . . . .	56
7.4	Semantics of data-presentations . . . . .	60
7.4.1	General . . . . .	60
7.4.2	The derivation system of a data-presentation . . . . .	60
7.4.2.1	Axioms generated by equations . . . . .	60
7.4.2.2	Inference rules generated by equations . . . . .	60
7.4.2.3	Generated derivation system . . . . .	60
7.4.3	Congruence relation induced by a data-presentation . . . . .	61
7.4.4	Quotient term algebra . . . . .	61
7.5	Semantics of a canonical LOTOS specification . . . . .	61
7.5.1	General . . . . .	61
7.5.2	Auxiliary definitions . . . . .	61
7.5.2.1	Notation . . . . .	61
7.5.2.2	Extended behaviour-expressions . . . . .	62
7.5.2.3	The simplification of sum- and par-expressions . . . . .	62
7.5.2.4	Substitution . . . . .	63
7.5.3	Transition derivation system . . . . .	63
7.5.3.1	General framework . . . . .	63
7.5.3.2	Axioms of transition . . . . .	63
7.5.3.3	Inference rules of transition . . . . .	64
7.5.4	Structured labelled transition system of a behaviour-expression . . . . .	69
7.5.4.1	Derivatives of a behaviour-expression . . . . .	69
7.5.4.2	Structured labelled transition system . . . . .	69
7.5.5	Formal interpretation of a canonical LOTOS specification . . . . .	70

## ANNEXES

A	Standard library of data types . . . . .	71
A.1	Introduction . . . . .	71
A.2	Syntax of the data type library . . . . .	71
A.3	Semantics of the data type library . . . . .	71
A.4	The Boolean data type . . . . .	72
A.5	Parameterized data type definitions . . . . .	73
A.5.1	Element . . . . .	73
A.5.2	Set . . . . .	74
A.5.3	Strings . . . . .	76
A.5.3.1	Non-empty string . . . . .	76
A.5.3.2	String . . . . .	78
A.6	Unparameterized data type definitions . . . . .	79
A.6.1	Natural number . . . . .	79
A.6.1.1	Abstract definition of natural numbers . . . . .	79
A.6.1.2	Representations of natural numbers . . . . .	80
A.6.1.2.1	Hexadecimal representation . . . . .	80
A.6.1.2.2	Decimal representation . . . . .	82

A.6.1.2.3	Octal representation . . . . .	83
A.6.1.2.4	Binary representation . . . . .	84
A.6.2	Octet . . . . .	85
A.6.3	Octet string . . . . .	85
<b>B</b>	<b>Equivalence relations . . . . .</b>	<b>87</b>
B.1	Introduction . . . . .	87
B.2	Weak bisimulation . . . . .	88
B.2.1	Definitions . . . . .	88
B.2.2	Laws for weak bisimulation congruence . . . . .	89
B.2.3	Laws for weak bisimulation equivalence . . . . .	92
B.2.3.1	Notation . . . . .	92
B.2.3.2	General law . . . . .	92
B.2.3.3	Rules for $\approx$ . . . . .	92
B.3	Testing equivalence . . . . .	92
B.3.1	Definitions . . . . .	92
B.3.2	Laws for testing congruence . . . . .	93
B.3.3	Laws for testing equivalence . . . . .	93
B.4	References . . . . .	94
<b>C</b>	<b>A tutorial on LOTOS . . . . .</b>	<b>95</b>
C.1	The specification of processes . . . . .	95
C.2	Behaviour expressions in basic LOTOS . . . . .	96
C.2.1	A basic process: inaction . . . . .	96
C.2.2	Two basic operators . . . . .	96
C.2.2.1	Action prefix . . . . .	96
C.2.2.2	Choice . . . . .	96
C.2.2.3	Processes as trees . . . . .	97
C.2.3	Recursion . . . . .	98
C.2.4	Parallelism . . . . .	99
C.2.4.1	Parallelism of independent processes . . . . .	99
C.2.4.2	Parallelism of dependent processes . . . . .	99
C.2.4.3	The general parallel operator . . . . .	100
C.2.4.4	The hiding operator . . . . .	101
C.2.4.5	Reasons for the hiding operator . . . . .	102
C.2.5	Nondeterminism in LOTOS . . . . .	103
C.2.6	Sequential composition of processes . . . . .	104
C.2.7	Disruption of processes . . . . .	105
C.2.8	An example in basic LOTOS . . . . .	106
C.3	LOTOS data types . . . . .	107
C.3.1	Introduction . . . . .	107
C.3.1.1	Basic concepts . . . . .	107
C.3.1.2	Abstract Data Types versus Concrete Data Types . . . . .	107
C.3.2	Concepts of LOTOS data types . . . . .	108
C.3.2.1	The signature . . . . .	108
C.3.2.2	Terms and equations . . . . .	108
C.3.2.3	The combination . . . . .	110
C.3.2.4	The parameterization . . . . .	111
C.3.2.5	Renaming . . . . .	113
C.3.2.6	Library invocation . . . . .	114
C.4	LOTOS with structured interactions . . . . .	114
C.4.1	Structured event offers . . . . .	114
C.4.1.1	Value declarations . . . . .	114
C.4.1.2	Variable declarations . . . . .	115
C.4.1.3	Types of interaction . . . . .	115
C.4.2	Conditional constructs . . . . .	116
C.4.2.1	Selection predicates . . . . .	116

C.4.2.2	Guarded expressions . . . . .	117
C.4.3	Process abstraction with parameterization . . . . .	117
C.4.4	Generalized choice and parallel expressions . . . . .	118
C.4.5	Generalized sequential composition . . . . .	119
C.4.5.1	Successful termination and functionality . . . . .	120
C.4.5.2	Functionality of LOTOS behaviour expressions . . . . .	120
C.4.5.3	Process abstraction and functionality . . . . .	121
C.4.5.4	Sequential composition with value passing . . . . .	122
C.4.5.5	Local variable definition . . . . .	122
C.4.6	Another example in LOTOS . . . . .	122
C.5	LOTOS syntax table . . . . .	123
D	Syntax diagrams . . . . .	127
E	Informal basis for abstract data types . . . . .	135
E.1	Introduction . . . . .	136
E.1.1	Representations . . . . .	136
E.2	Signatures . . . . .	138
E.3	Terms and expressions . . . . .	139
E.3.1	Generation of terms . . . . .	139
E.4	Values and algebras . . . . .	140
E.4.1	Equations and quantification . . . . .	141
E.5	Algebraic specification and semantics . . . . .	141
E.6	Representation of values . . . . .	142

## TABLES

1	Metalanguage symbols . . . . .	8
2	Actualization function . . . . .	43
3	Functional structure of flattening function . . . . .	56
4	Interaction types . . . . .	116
5	LOTOS syntax table . . . . .	123

## FIGURES

1	Structure of clause 7 . . . . .	26
2	Two interacting processes . . . . .	95
3	Full duplex buffer . . . . .	96
4	Composition of two buffer processes . . . . .	100
5	Hiding . . . . .	101

This page intentionally left blank



---

# Information processing systems — Open Systems Interconnection — LOTOS — A formal description technique based on the temporal ordering of observational behaviour

## 0 Introduction

### 0.1 General

Formal description techniques (FDTs) are methods of defining the behaviour of an (information processing) system in a language with a formal syntax and semantics, instead of a natural language such as English. In the following sub-clauses of this introduction, the importance of FDTs and their standardization is discussed. The objectives that an FDT must satisfy are considered. The origin of LOTOS is discussed. Finally the structure of this document is explained.

### 0.2 FDTs

Formal description techniques are important tools for the design, analysis and specification of information processing systems. It is by means of formal techniques that system descriptions can be produced that are *complete, consistent, concise, unambiguous* and *precise*. This is only possible if an FDT is self-contained, so that the descriptions given in an FDT need not refer to any informal knowledge of the system that is described. An important aspect of a formal system is that it allows analysis by mathematical methods. An FDT that has such a formal, mathematical basis can be used to prove the correctness of specifications.

### 0.3 The requirement for standard FDTs

If an FDT is defined in an International Standard, the description is available to all who require it. The Directives for the production of such a standard require a high degree of international acceptance and technical stability. Any amendment also requires international agreement. Hence a standard FDT offers the most useful form of presentation to those who wish to apply it.

### 0.4 The objectives to be satisfied by an FDT

Although this document describes an FDT that is generally applicable to distributed, concurrent information processing systems, it has been developed particularly for OSI. The main objectives to be satisfied by such an FDT are that it should be

- a) *expressive*: an FDT should be able to define both the protocol specifications and the service definitions of the seven layers of OSI described in ISO 7498.
- b) *well-defined*: an FDT should have a formal mathematical model that is suitable for the analysis of these specifications and definitions. The same model should support the checking of conformance of implementations that are permitted by the OSI International Standards. This model should also support the testing of an implementation for conformance.
- c) *well-structured*: an FDT should offer means for structuring the description of a specification or definition in manner that is meaningful and intuitively pleasing. A good structure increases the readability, understandability, flexibility, and maintainability of system descriptions, and offers a better framework for their analysis.
- d) *abstract*: there are two aspects of abstraction that an FDT should offer:
  - 1) an FDT should be completely independent of methods of implementation, so that the technique itself does not provide any undue constraints on implementors
  - 2) an FDT should offer the means of abstraction from irrelevant details with respect to the context at any point in a description. Abstraction can reduce the local complexity of system descriptions considerably. In the presence of a good structure, abstraction can help even further to reduce the complexity of descriptions.

## 0.5 The origin of LOTOS

LOTOS (Language of Temporal Ordering Specification) was developed by FDT experts from ISO/TC97 during the years 1981-1988. The basic idea that LOTOS developed from was that systems can be described by defining the temporal relation between events in the externally observable behaviour of a system. LOTOS has two components. The first component deals with the description of process behaviours and interactions, and is based on a modification of the Calculus of Communicating Systems (CCS), which was developed at the University of Edinburgh. The modification includes elements that were introduced in other calculi, which are related to CCS, viz. CSP and CIRCAL. Among the other theories that are related to CCS, and thus to LOTOS, are SCCS, MEIJE and ACP. CCS, and the related formal systems, provide a powerful analytical theory for concurrent processes.

The second component deals with the description of data structures and value expressions and is based on the abstract data type language ACT ONE. ACT ONE was developed at the Technical University of Berlin. The part of LOTOS dealing with the description of processes, i.e. dynamic behaviours, is not dependent upon ACT ONE. Many well-defined languages for the description of data structures could, in principle, be used in combination with the process definition facilities of LOTOS.

## 0.6 The structure of this International Standard

This document differs in contents from most International Standards. The importance of a formal, mathematical basis of an FDT (see clauses 0.2 and 0.4 b)) makes the inclusion of mathematical material in this definition necessary. Clause 4 introduces some fundamental mathematical concepts and notations that are used in the rest of the document. Clause 5 presents the fundamental mathematical structures that provide a semantic basis for LOTOS data types, behaviour expressions, and their combination. Clause 6 presents the syntax of the language and contains, together with 7.3, the rules for producing syntactically correct specifications; this part of the document requires knowledge of only basic mathematical concepts. Clause 7 presents the semantics of a LOTOS specification, based on the semantics of data types and behaviour expressions. Annex A contains the standard library of LOTOS data types and forms a part of this International Standard. The other annexes provide more information related to LOTOS, but do not form a part of the standard. In particular, annex C contains a *tutorial* on LOTOS, which is meant to provide a guide to the features of the language, and a convenient introduction to this standard for the non-technical reader.

## 1 Scope and field of application

This International Standard defines the syntax and semantics of the Formal Description Technique LOTOS. LOTOS is in general used for the formal description of distributed, concurrent information processing systems. In particular LOTOS can be used to describe formally the service definitions and protocol specifications of the layers of Open Systems Interconnection (OSI) architecture described in ISO 7498, and related standards, and conformance tests for implementations of OSI protocols and/or OSI functions. It can also be applied for the formal description of other distributed systems, such as telephone switching networks.

## 2 References

ISO 7498, *Information processing systems - Open Systems Interconnection - Basic Reference Model*.

CCITT Recommendation Z.100, *SDL*.