



中华人民共和国国家标准

GB/T 16656.11—1996

工业自动化系统与集成 产品数据表达 和交换 第 11 部分:描述方法: EXPRESS 语言参考手册

Industrial automation systems and integration—
Product Data Representation and Exchange—
(STEP)

Part 11: Descriptive methods:
The EXPRESS language reference manual

1996-12-17 发布

1997-07-01 实施

国家技术监督局 发布

目 次

1 主题内容与适用范围	1
2 引用标准	1
3 术语	2
4 一致性要求	2
5 基本原理	3
6 语言规范语法	4
7 基本语言元素	5
8 数据类型	11
9 说明	20
10 作用域和可见度	40
11 接口规范	47
12 表达式	50
13 可执行语句	66
14 内部常数	72
15 内部函数	73
16 内部过程	83
附录 A EXPRESS 语言语法(参考件)	85
附录 B 对子类型/超类型图的实体示例定义(参考件)	93
附录 C 由接口规范所加的引例限制(参考件)	102
附录 D EXPRESS-G;EXPRESS 语言的图形子集(参考件)	103
附录 E 协议实现的一致性语句(参考件)	112
附录 F 说明 EXPRESS-G 图的 EXPRESS 模型示例(参考件)	113

中华人民共和国国家标准

工业自动化系统与集成 产品数据表达 和交换 第 11 部分:描述方法: EXPRESS 语言参考手册

GB/T 16656.11—1996

Industrial automation systems and integration—
Product Data Representation and Exchange—
(STEP)

Part 11: Descriptive methods:
The EXPRESS language reference manual

本标准等效采用 ISO DIS 10303—11:1994《工业自动化系统与集成—产品数据表达和交换—第 11 部分:描述方法:EXPRESS 语言参考手册》。

1 主题内容与适用范围

1.1 主题内容

本标准是在《工业自动化系统与集成—产品数据表达和交换》总标题下系列标准 GB/T 16656 之一。

GB/T 16656 规定了产品信息的表示以及所需的机制与定义,以便在与产品设计、制造和其他应用相关的不同计算机系统或环境间进行数据交换。

GB/T 16656 是一个系列标准,它由综述和基本原理、产品描述方法、集成通用资源、集成应用资源、应用协议、实现方法以及一致性测试和框架七个类别组成。EXPRESS 语言参考手册是产品描述方法类中的一个标准。

GB/T 16656.11 规定了一种可用于描述产品数据特性的语言。这种语言称作 EXPRESS 语言,它是一种数据规范语言,由用于一种无歧义对象定义的语言元素和对定义对象限制的规范所组成。

1.2 适用范围

本标准适用于对产品数据表达和交换的描述方法的规定,即使用本语言对 STEP 系列标准中的模式、实体、约束条件及属性进行描述。

它的范围包括数据类型以及对数据类型引例的限制。

它不能用于程序编制;

它不能用于规定数据库格式、文件格式和转换格式的定义;

它也不能用于过程控制,信息处理和异常处理。

2 引用标准

GB 13000.1—93 信息技术 通用多八位编码字符集(VCS)第一部分:体系结构与基本多文种平面
ISO 10303—1 工业自动化系统与集成 产品数据表达和交换 概述和基本原理

国家技术监督局 1996-12-17 批准

1997-07-01 实施

3 术语

3.1 复合实体数据类型 complex entity data type

在特定的子类型/超类型图中所许可的实体数据类型组合。

3.2 复合实体引例 complex entity instance

复合实体数据类型的引例。

3.3 常数 constant

来自一个不能被修改的特定域的命名值。

3.4 数据类型 data type

引例的范围。

3.5 实体 entity

具有共同性质的一类对象。

3.6 实体引例 entity instance

对象的表示,这种表示具有与特定实体数据类型的公共性质相连的值。实体引例分为两类:简单实体引例和复合实体引例。

3.7 引例 instance

对象的表示。

3.8 对象 object

在现实世界中存在的概念或物理事物。

3.9 局部复合实体数据类型 partial complex entity data type

在可构成部分复合实体数据类型的子类型/超类型图内的一组实体数据类型。

3.10 局部复合实体数据引例 partial complex entity data instance

局部复合实体数据类型的引例。它本身无意义,它与其它局部复合实体引例一起构成复合实体引例。

3.11 总体 population

引例的集合。

3.12 性质 property

用属性或限制表示的现实世界特征。

3.13 简单实体引例 simple entity instance

简单实体数据类型的引例。

3.14 子类型/超类型图 subtype/supertype graph

已说明的实体数据类型集合,在子类型/超类型图中被说明的实体。

数据类型通过子类型语句相关。一个子类型/超类型图定义一个或多个复合实体数据类型。

3.15 记号 token

语言的不可分解的词法元素。

4 一致性要求

4.1 用 EXPRESS 语言书写的形式规范

4.1.1 词法语言

按 EXPRESS 语言书写的形式规范应与下述规定的所给定的检查层次相一致。支持某个给定层次用的 EXPRESS 语言写的形式规范也应当支持低于该层次的所有层次。

4.1.1.1 检查的层次

第一层(Level 1):引用检查。该层检查形式规范的组成应可确保形式规范能用附录 A 中已扩展的 EXPRESS 语言的基本语法规则(模式说明)生成,它提供了一个全面的语法检查,或缺省为一个全面的

引用检查。

第二层(Level 2):类型检查。该层检查形式规范的组成应可标识出那些在表达式中类型兼容(见第12.12节)和赋值兼容(见第13.3条)无效的地方。

第三层(Level 3):值检查。该层检查形式规范的组成应可标识出值可被求解的地方,即从第7章到第16章中所规定的'A应大于B'这种语句不一致的那些地方。

第四层(Level 4):全面检查。该层中检查形式规范的组成应可标识出那些满足一致性规范(在本标准规定)的语句未被填满的地方。

例:

本标准规定,当一个函数被调用时,在每一个可取的路径上应规定一个返回语句。这种规定就应当得到检查。

4.1.2 图的形式

用 EXPRESS-G 图书写的形式规范应与下面规定的给定检查层次相一致,支持给定层次的用 EXPRESS-G 图写的形式规范也应当支持低于该层次的所有层次。

4.1.2.1 检查的层次

第一层(Level 1):符号检查。该层检查形式规范的组成应可标识出那些与在附录 D3 和附录 D4 中分别规定的实体层或模式层规范不一致的地方,它也包括对形式规范中使用附录 D2、附录 D3 和附录 D4 中规定符号的检查。

第二层(Level 2):全面检查。该层检查形式规范的组成应可标识出那些与在附录 D 中定义的完整实体或完整模式层规范和在第7章到第16章中所定义的要求不一致的地方。

4.2 EXPRESS 语言的实现方法

4.2.1 EXPRESS 语言的分析/检验程序

EXPRESS 语言语法分析程序的实现方法应对用 EXPRESS 语言写的并与附录 E 中为该实现方法所说明的限制相一致。如果 EXPRESS 语言语法分析程序可按给定层次(包括比该层次低的任何层次)要求对用 EXPRESS 语言写的形式规范实行所有检查,则可以认为该分析程序与这个特定的检查层(在 4.1.1.1 条中所定义的)是一致的。

EXPRESS 语言语法分析程序的实现方法应对标识符长度、处理数的范围和实数的最高精度的限制作出规定。为一致性测试的目的,这些限制应按附录 E 规定的格式编制成文件。

4.2.2 图形编辑工具

EXPRESS-G 编辑工具的实现方法应能生成用 EXPRESS-G 表示的形式规范,该形式规范应与在附录 E 中为该实现方法所规定的限制相一致。

如果 EXPRESS-G 编辑工具所生成的图能与在使用附录 D2~附录 D4 中规定的符号所生成的实体层图或模式层图相一致,则可以认为 EXPRESS-G 编辑工具与符号层(如第 4.2.2.1 条所定义的)是一致的。

如果 EXPRESS-G 编辑工具所生成的图能与使用在附录 D2~附录 D4 中规定的符号所生成的完整实体层图或完整模式层图一致,且按附录 D7 中规定要求实现检查,则可以认为 EXPRESS-G 编辑工具与全面检查层(如第 4.1.2.1 条所定义的)是一致的。

EXPRESS-G 编辑工具的实现方法应对数、标识符长度、每页模型的对象数和最大页数的限制作出规定。为一致性测试的目的,这些限制应按附录 E 规定的格式编制成文件。

5 基本原理

本标准是以下述概念为前题的。

用 EXPRESS 语言写的模式描述建立一个域的一组条件。为确定数据引例是否在该域内,可求解数据引例。如果这些数据引例满足所有条件,则可以认定这些数据引例是在该域中。如果这些引例不能满

足任何条件,则这些数据引例是违背这些条件的,且因此是不在这个域中。如果数据引例没有包含可选属性的值,而一些条件与那些可选属性有关,这时就可能不能确定这些数据引例是否满足所有条件,在这种情况下,这些数据引例是被看作在域中。

在 EXPRESS 语言中的实体数据类型规范描述一个域,该域中的单个成员由唯一的相关标识符区分开。EXPRESS 语言不规定这些标识符的内容和表示。

EXPRESS 语言的许多元素赋值有名字,这些名字使其它语言元素能引用这些相关的表示。在其它语言元素定义中的名字使用构成了对下划线表示(underlying representation)的引用。当该语言的语法对该名字使用一个标识符时,为理解这种结构,应对这种下划线表示进行解释。

EXPRESS 语言不描述实现方法的环境,特别是 EXPRESS 不规定下述事项:

- 对名字的引用是如何解决的;
- 其它模式是如何知道的;
- 限制是在何时或如何检查的;
- 不管与 EXPRESS 模式不一致的引例是否允许在实现方法中存在。

6 语言规范语法

用来表示 EXPRESS 语言语法的表示法在本章定义。

EXPRESS 语言中的所有语法在附录 A 中给出。为说明各种语句的语法,这些语法规则的各部分将在以后各章节中分别阐述。各个章节中的叙述不总是完整的,因此有时对所缺少规则部分要参阅附录 A。在本标准中语法部分的每条规则都有一个唯一的号排在每个规则的左侧,以便于其它语法规则的交叉引用。

6.1 语法规范

EXPRESS 语言语法是按沃思语法表示法(Wirth Syntax Notation)方案定义的。

本标准中定义的表示法约定和沃思语法表示法给出如下:

语法 = {产生式}.

产生式 = 标识符 '=' 表达式'.'

表达式 = 项{'|'}项.

项 = 因子{因子}.

因子 = 标识符 | 文字 | 组 | 选项 | 重复.

文字 = '''字符{字符}'''.

组 = '('表达式')'.

选项 = '['表达式']'.

重复 = '{'表达式}'.

等号 '=' 表示一个产生式,左边的元素定义为右边元素的结合。除了出现在文字内的空格外,任何出现在产生式元素间的空格都是无意义的。一个产生式用一个句号 '.' 结束。

标识符的引例用来表示出现在另一个产生式左边的非终结符号。标识符引例由字母、数字和下划线字符组成。语言的关键字由只用大写字母给出的标识符的产生式表示。

词'文字'用来表示不能再扩展的终结符号,文字是括在撇号中的状态独立的字符序列。在这种情况下,字符表示由 GB 13000.1 在 00 组 00 平面 00 行中的 21-7E 位所定义的任何一个字符。出现在文字中的撇号,应当写两次。各种括号的语义定义如下:

花括号 '{ }' 表示零次或多次重复;

方括号 '[']' 表示可选参数;

圆括号 '(')' 表示由括号括起来的产生式组作为单个产生式使用;

竖线 '|' 表示逻辑异或。

例：

串类型语法如下：

语法：

286 字符串类型=STRING['('宽度')'][FIXED].

306 宽度=数值表达式.

完整的语法定义(见附录 A)包括对串、数值表达式和域的定义。按上面给定的语法,下面的替换是允许的:

```
string
string(22)
string(19)fixed
```

6.2 专用字符说明

下面的说明用来表示整个字符集和显示较困难的专用字符:

\a 表示 GB 13000.1 中的 00 组、00 平面、00 行、21-7E 位中的字符;

\n 表示一个新行(系统相关的)(见第 7.1.5.2 条);

\q 引号(撇号)(')字符,并且包括在符号\a 内;

\s 空格字符;

\o 表示在 GB 13000.1 中 00 组、00 平面、00 行、00~1F 位中的字符。

7 基本语言元素

本章规定构成 EXPRESS 模式的基本元素、字符集、注解、符号、保留字标识符和文字。

本语法定义引自 EXPRESS 语言语法(见附录 A),该语法的表示方法在第六章定义。

基本语言元素由一系列正文构成,一般为几个物理行。每个物理行由新行结束的任意个(也可以是零个)字符组成(见第 7.1.5.2 条)。下面两个例子是等效的。

例:

```
entity point;x,y,z:real;end entity;
```

例:

```
ENTITY point;
x;
y;
z:REAL;
END_ENTITY;
```

7.1 字符集

用 EXPRESS 语言书写的模式只使用由下述选择的 GB 13000.1 子集定义的字符:00 组、00 平面、00 行的 00-7F 位。这个选择的 GB 13000.1 子集的可打印字符的组合形成 EXPRESS 语言的记号,这些记号包括关键字、标识符、符号或文字。

注:本章只涉及用于规定 EXPRESS 模式的字符,而不规定在串数据类型内所允许的字符域。

7.1.1 数字

语法:

118 数字='0'|'1'|'2'|'3'|'4'|'5'|'6'|'7'|'8'|'9'.

EXPRESS 语言使用阿拉伯数字 0~9(EXPRESS 字符集的第 30~39 位)。

7.1.2 字母

语法:

122 字母='a'|'b'|'c'|'d'|'e'|'f'|'g'|'h'|'i'|'j'|'k'|'l'|'m'|'n'|'o'|'p'|'q'|'r'|

's' | 't' | 'u' | 'v' | 'w' | 'x' | 'y' | 'z'.

EXPRESS 语言使用英文字母串表中的大写和小写字母(EXPRESS 字符集第 41~5A 和 61~7A 位),只在显示串文字内字母的大小写才有不同的意义。

注: EXPRESS 语言可用大写、小写或混合字母书写。

7.1.3 下划线

下划线(_ ,EXPRESS 字符集的第 5F 位)可用在标识符和关键字中,但它不能作为第一个字符使用。

7.1.4 专用字符

语法:

132 专用字符=非括号、星号、引号字符|' * '|'('|')'|''''.

126 非括号、星号、引号字符='!'|'@'|'#'|'\$'|'%'|'&'|'&'|'-'|'-'|'+'|'='|'
'{|'|'}'|'['|']'|'~'|':'|';'|'|'""|'""|'<'|'>'|','|'.|'|'
'?'|'|'/'|'|'\'.

专用字符主要用来作为标点符和操作符。某些专用字符不作为语言的一部分使用,它们可以用在注解和显式串文字中。这些专用字符是 EXPRESS 字符集的第 21~2F。

3A~3F、40、5B~5E、60 和 7B~7E 位。

7.1.5 空白

空白一般用来分隔记号,且常用于改善结构,使得所写语言易读,即缩排说明或留出空行。

7.1.5.1 空格符

在两个记号之间或在一个显式串文字中可以出现一个或多个空格符(EXPRESS 字符集的第 20 位)。在语言的语法中,\s 表示空格符。

7.1.5.2 新行

新行在用 EXPRESS 书写的形式规范中表示一行的物理结束。新行在正常情况下作为空格处理,只有当它终结一个尾部注解或者异常地终结一个串文字时,它才有实际意义。在语言的语法中,新行用\n 表示。新行的表示是实现方法定义的。

注:新行表示可是下述字符之一:专用字符、字符组合、伪字符或信号。

7.1.5.3 其它字符

在第 7.1.1 节和 7.1.5.2 条中没有定义的字符(这些字符是 EXPRESS 字符集的第 00~1F 和 7F 位),除非是在显式串文字中出现的,否则将作为空白处理,本语言语法规定,用\0 表示其它字符。

7.1.6 注解

注解用于编制文件,只有作空白用时才有意义。注解有两种形式,可以放在两个记号之间的是嵌入注解。尾部注解出现在一个物理行结束处,且与行的终结符等效。

7.1.6.1 嵌入式注解

字符对(* 标明一个嵌入式注解的开始,而字符对*)标明它的结束。嵌入式注解可出现在任何两个记号之间。

语法:

140 嵌入式注解='(* '{非左括号星号|左括号非星号|星号非右括号|嵌入式注解}' *)'.

124 非左括号星号=非括号星号|')'.

125 非括号星号=字母|数字|非括号星号专用.

127 非括号星号专用=非括号星号引号专用|''''.

126 非括号星号引号专用='!'|'@'|'#'|'\$'|'%'|'&'|'&'|'-'|'-'|'+'|'='|'{'|'|'}'|'
['|']'|'~'|':'|';'|'|'""|'""|'<'|'>'|','|'.|'|'?'|'|'/'|'|'\'.

118 数字='0'|'1'|'2'|'3'|'4'|'5'|'6'|'7'|'8'|'9'.

123 左括号非星号='('非星号.

130 非星号=非括号星号|'('|')'.

133 星号非右括号='*'|'('.

129 非右括号=非括号星号|'*'|'('.

EXPRESS 语言选择子集中的任何字符可出现在星号和包括新字符在内的嵌入式注解之间,因此嵌入式注解可以占多个物理行。

嵌入式注解可以嵌套。

例:下面为嵌套的嵌入式注解的例子。

(* The'(* 'symbol starts a remark and the'*)'symbol end it *)

注:嵌套式注解时要注意确保符号的配对。

7.1.6.2 尾部注解

语法:

142 尾部注解='--'{'a|\s|\o'}\n.

尾部注解必须写在一个物理行末尾,一个尾部注解用两个连续的短横连字符(--开始,以一个新行结束。

例:

--this is a remark that end with a newline

7.2 符号

符号是在 EXPRESS 语言中有专门意义的字符或字符组,EXPRESS 语言中使用的符号作为分隔符或操作符。分隔符用来分隔相邻的语法元素,否则词法元素的解释成为不可能;操作符则完成与操作符有关的运算项的运算。第 12 章将对操作符进行全面的阐述。表 1 中列出了 EXPRESS 的符号。

表 1 EXPRESS 符号

.	,	;	:
*	+	-	=
\	/	<	>
[]	{	}
	()	**
<=	<>)=	<*
:=		:=:	:(>:
%	,	(*	*)
--	e		

7.3 保留字

EXPRESS 的保留字是关键字和内部常数、函数和过程的名字。保留字不能用作标识符。

7.3.1 关键字

EXPRESS 的关键字见表 2。

表 2 EXPRESS 关键字

ABSTRACT	AGGREGATE	ALIAS	ARRAY
AS	BAG	BEGIN	BINARY
BOOLEAN	BY	CASE	CONSTANT
CONTEXT	DERIVE	ELSE	END
END ALIAS	END CASE	END CONSTANT	END CONTEXT
END ENTITY	END FUNCTION	END IF	END LOCAL
END MODEL	END PROCEDURE	END REPEAT	END RULE
END SCHEMA	END TYPE	ENTITY	ENUMERATION
ESCAPE	FIXED	FOR	FROM
FUNCTION	GENERIC	IF	INTEGER
INVERSE	LIST	LOCAL	LOGICAL
MODEL	NUMBER	OF	ONEOF
OPTIONAL	OTHERWISE	PROCEDURE	QUERY
REAL	REFERENCE	REPEAT	RETURN
RULE	SCHEMA	SELECT	SET
SKIP	STRING	SUBTYPE	SUPERTYPE
THEN	TO	TYPE	UNIQUE
UNTIL	USE	VAR	WHERE
WHILE			

注：① 关键字的表示文字的大写产生式使语法产生式的阅读变得容易。

② 关键字 CONTEXT、END CONTEXT、END MODEL 和 MODEL 保留为本标准以后使用。

7.3.2 作为操作符的保留字

表 3 中给出了由保留字定义的操作符。

表 3 EXPRESS 操作符保留字

AND	ANDOR	DIV	IN
LIKE	MOD	NOT	OR
XOR			

7.3.3 内部常数

表 4 中给出了内部常数名,这些常数的定义见第 14 章。

表 4 EXPRESS 内部常数保留字

?	SELE	CONST_E	PI
FALSE	TRUE	UNDOWN	

7.3.4 内部函数

表 5 中给出了内部函数的名字,这些函数的完整定义见第 15 章。

表 5 EXPRESS 函数名保留字

ABS	ACOS	ASIN	ATAN
BLENOTH	COS	EXISTS	EXP
FORMAT	HIBOUND	HINDEX	LENGTH
LOBOUND	LOG	LOG2	LOG10
LOINDEX	NVL	OOD	ROLESOF
SIN	SIZEOF	SQRT	TAN
TYPEOF	USEDIN	VALUE	

7.3.5 内部过程

表 6 中给出了内部过程名,这些内部过程的完整定义见第 16 章。

表 6 EXPRESS 过程名保留字

INSERT	REMOVE
--------	--------

7.4 标识符:

语法:

138 简单标识符=字母{字母|数字|'-'}

118 数字='0'|'1'|'2'|'3'|'4'|'5'|'6'|'7'|'8'|'9'.

标识符是在模式中给说明项(包括模式本身)定的名字,标识符不应与 EXPRESS 中的保留字相同。

标识符的第一个字符必须是字母,其余的可以是字母、数字或下划线的组合。

EXPRESS 语言语法分析程序的实现应按附录 E 规定该实现方法所能读的标识符字符的最大个数。

7.5 文字

语法:

231 文字=二进制文字|整数文字|逻辑文字|实数文字|串文字。

文字是自定义的常数值,文字的类型取决于如何组合字符成为记号,文字类型可以是二进制型、整数型、实数型、串型或逻辑型。

7.5.1 二进制文字

语法:

134 二进制文字='%'位{位}.

117 位='0'|'1'.

二进制文字表示二进制数据类型的值,由表示二进制文字的符号%和其后跟的一个或多个位(0 或 1)组成。

EXPRESS 语言语法分析程序的实现应按附录 E 规定该实现方法所能读的二进制文字的最大位数。

例:有效的二进制文字:

%0101001100

7.5.2 整数文字

整数文字表示一个整数数据类型的值,它由一个或多个数字组成。

语法:

136 整数文字=数字.

119 数字=数字{数字}.

118 数字='0'|'1'|'2'|'3'|'4'|'5'|'6'|'7'|'8'|'9'.

EXPRESS 语言语法分析程序的实现方法应按附录 E 规定该实现所能读的整数文字的最大整数值。

例:有效的整数文字:

4016

38

7.5.3 实数文字

实数文字表示实数类型的值,它由一个数值部分和一个任选的指数部分组成,数值部分应包括所有小数点。

语法:

137 实数文字=数字'·'[数字][‘e’][符号][数字].

119 数字=数字{数字}.

118 数字='0'|'1'|'2'|'3'|'4'|'5'|'6'|'7'|'8'|'9'.

279 符号='+'|'-'.

例:有效的实数文字:

1. E6 “E”可以用大写,也可以用小写。

3. 5e-5

359. 62

例:无效的实数文字:

.001 小数点前至少有一个数字。

1e10 小数点应是文字的一部分。

1. e10 空格不是实数文字的一部分。

7.5.4 串文字

串文字表示串数据类型的值。串文字有两种类型:显式串文字和编码串文字。显式串文字由引号(‘)括起来的 EXPRESS 字符集中的字符系列组成(第 7.1 节),显式串文字内的引号用两个连续引号表示,编码串文字为用双引号(”)括起来的在 GB 13000.1 中的四个八位位组编码表示的字符序列。

串文字不应超越一个物理行边界,即在括起来串文字的引号之间不应出现新行。

语法:

285 串文字=简单串文字|编码串文字.

139 简单串文字=\q{(\q\q)|非引号|\s|\o}\q.

128 非引号=非引号星号引号专用|字母|数字|‘(‘|’|’|*’.

126 非括号星号引号专用='!'|'”|'#'|'\$'|'%'|'&'|'+'|','|'-'|'·'|'|/'|':'|';'|'|<|'|='|'|>|'|?|'|@|'|[|'|\\'|']|'|^'|'|_'|'|“|'|{ '|'|}|'|} '|'|~'.

128 数字='0'|'1'|'2'|'3'|'4'|'5'|'6'|'7'|'8'|'9'.

135 编码串文字='”编码字符{编码字符}””.

120 编码字符=八位位组 八位位组 八位位组 八位位组.

131 八位位组=六位数字 六位数字.

121 六位数字=数字|'a'|'b'|'c'|'d'|'e'|'f'.

EXPRESS 语言语法分析程序的实现应按附录 E 规定该实现方法所能读的显式串文字最大字符数。

EXPRESS 语言语法分析程序的实现也应按附录 E 规定该实现方法所能读的编码串文字的最大 8 位位组数(它应是 4 的倍数)。

例:有效的显式串文字:

'Baby needs a new pair of shoes!'

读作:孩子需要一双新鞋!

'ED' 'S computer store'

读作:伊迪的计算机商店。

例:无效的显式串文字:

'ED'S computer store'

引号应总是成对出现。

'ED'S computer store'

Store'

超过了一个物理行。

例:有效的编码串文字:

"00000041"读作 A。

"000000c5"读作 Å。

例:无效的编码串文字:

"000041"

8 位位组应是 4 个一组。

"00000041 000000c5"

""之间仅允许有六个字符。

7.5.5 逻辑文字

逻辑文字表示逻辑数据类型或布尔数据类型的值,它是内部常数真(TRUE)、假(FALSE)和未知(Unknown)之一。

语法:

235 逻辑文字=FALSE|TRUE|UNKNOWN.

注:未知与布尔数据类型不兼容。

8 数据类型

本章定义该语言所提供的数据类型,数据类型定义一个引例域,每个属性局部变量或者形参都有一个定义它的域的相关数据类型。

数据类型可以分为简单数据类型、聚合数据类型、命名数据类型、构造数据(constructed data)和广义数据(generalized data)类型。数据类型也可按它们的应用分为基本数据类型、参数数据类型和下划线数据类型。这两种分类间的关系见第 8.6 节。

对这些数据类型所能实施的操作在第 12 章定义。

8.1 简单数据类型

简单数据类型定义 EXPRESS 中的最小数据单元的域,也就是说它们不能再细分为 EXPRESS 所标识的元素。简单数据类型分为数值型、实数型、整数型、串型、布尔型、逻辑型和二进制型。

8.1.1 数值型数据类型

数值型(NUMBER)数据类型以本语言的所有数字的值,无论是整数、实数,作为它的域。在那些不必采用更专门的数值表示的场合,可以使用这种数据类型。

语法:

241 数值型=NUMBER.

例:

size:NUMBER;

因为我们可能不知道度量(SIZE)的使用场合,因此我们不知道如何正确表示它。

例如,足球比赛参加人数为整数,而球场的大小应为实数。

8.1.2 整数型数据类型

语法:

222 整数型=INTEGER.

整数型(INTEGER)数据类型以所有整数作为它的域,它的范围是在实数型数据类型中划定的一个限定范围。

例:本例用整数型表示名为节点的属性,这个属性的域是未加其它限制的所有整数。

```
ENTITY foo;
    nodes:INTEGER;
    ...
END_ENTITY;
```

8.1.3 实数型数据类型

语法:

258 实数型=REAL['('精度说明')'].

248 精度说明=数值表达式.

实数型(REAL)数据类型以所有有理数、无理数和科学计算实数作为它的域,它的范围是在数值型数据类型中划定的一个限定范围。

有理数和无理数有无限个解,并且是精确的。科学计算的数值只表示对给定精度为已知的值,精度说明用有效数字位表示。

实数文字由数值部分和任选指数部分表示。有效位数由把所有的前置零去掉时构成数值部分的数字个数表示。已知值的精度是前置数字的个数,而这个精度由应用或现有测量技术的限制而定。

规则和限制:

- a. 精度说明给出所求值的数位的最小个数,该表达式的值应为一个正整数解;
- b. 当值的精度说明未给出时,则实数的精度不受限制。

8.1.4 逻辑型数据类型

语法:

236 逻辑型=LOGICAL.

逻辑型(LOGICAL)数据类型以文字真(TRUE),假(FALSE)和未知(UNKNOWN)作为它的域。下述规则适于逻辑类型的值:FALSE<UNKNOWN<TRUE. 除未知的值不能赋值到布尔变量外,逻辑型与布尔型是兼容的。

8.1.5 布尔型数据类型

语法:

171 布尔型=BOOLEAN.

布尔型(BOOLEAN)数据类型以文字真(TRUE)或假(FALSE)作为它的域。

布尔型可以看作是对逻辑型的一种限定。下述规则也适于布尔型数据类型,FALSE<TRUE.

例:本例用布尔型数据类型表示一个命名为平面的(PLANAR)属性与平面的引例相连的值可为TRUE或FALSE。

```
ENTITY surface;
    planar:BOOLEAN;
END_ENTITY;
```

8.1.6 二进制型数据类型

语法:

170 二进制型=BINARY['('宽度')'][FIXED]].

306 宽度=数值表达式。

二进制型(BINARY)以每位为 0 或 1 的值的位序列作为它的域。

二进制可以定义为定长的,也可定义为可变长度的。如果二进制型未被说明为定长度的(即在定义中未使用 FIXED 保留字),则该二进制型为可变长度的。定长度二进制型数据类型的域为长度恰好是在类型定义中所规定长度(位数)的所有位序列的集合。

可变长度二进制型数据类型的域为长度小于或等于在类型定义中所规定的最大宽度的所有位序列集合。如果未规定宽度,则该域为对这些序列长度不加限制的所有位序列集合。

二进制数据类型的各个部分和单个的位可使用按第 12.3 节规定的下标寻址。

规则和限制:

- a. 宽度(任选项)给定所能容纳的最小位数,该表达式值应为一个正整数;
- b. 没有规定宽度时,二进制值的位数是不受本语言限制的。

例:本例可用来容纳字符的字形信息。

ENTITY character;

representation; ARRAY[1 : 20] OF BINARY(8) FIXED;

END ENTITY;

8.1.7 串型数据类型

语法:

286 串型=STRING['('宽度')'[FIXED]].

306 宽度=数值表达式。

串型 (STRING) 数据类型以字符序列域作为它的域。为串型数据类型定义的字符可以是 GB 13000.1 表示的任何字符。

串型数据类型可以定义为固定长度的或可变长度的。如果未规定为定长度的(即在定义中未使用 FIXED 保留字),则该串为可变长度的。

定长度串型数据类型为长度正好是在类型定义中所规定长度的所有字符序列的集合,可变长度串型数据类型为长度小于或等于在类型定义中所定义的最大宽度的所有字符长度序列的集合。

如果没有规定宽度,则该域为对这些字符长度不加限制的所有字符序列的集合。

子串和单个的字符可使用按第 12.5 节所规定的下标寻址。

在串内,字母的大小写都是有效的。

规则和限制:

- a. 宽度(任选项)给定所能容纳的最大字符个数,表达式值应为一个正整数;
- b. 当未规定宽度时,字符个数是不受本语言限制的。

例:

string1:STRING;

定义一个未规定最大长度的可变长度串。

string2:STRING(10);

定义一个最大长度为 10 个字符的串,但它的实际长度可在 0 到 10 个字符间变化。

string3:STRING(10) FIXED;

定义一个长度恰好为 10 个字符的串,其实际长度是不能改变的。

8.2 聚合数据类型

聚合数据类型以给定的基本数据类型引例集合作为它的域(见第 8.6.1 条),这些基本数据类型引例称为聚合引例的元素。EXPRESS 语言提供四种聚合数据类型:数组、数表、数袋、和数集。每种聚合数据类型有不同的性质和它的引例相连。

数组是一个有固定范围的有序集合,它由一个整数系列标引,该数据类型定义规定不限制一个引例