

CHILL语言定义

—CCITT建议书 Z·200(1984)

寇化栋 译

南京通信工程学院

一九八四年十二月

目 录

1 · 绪 论	1
1 · 1 概述	1
1 · 2 语言的总观	2
1 · 3 模式和类别	3
1 · 4 单元及其访问	4
1 · 5 值及其操作	5
1 · 6 动 作	6
1 · 7 输入和输出	7
1 · 8 程序结构	7
1 · 9 并发执行	8
1 · 10 总的语义性质	9
1 · 11 异常处理	10
1 · 12 实现任选	10
2 · 预备知识	12
2 · 1 元语言	12
2 · 1 · 1 上下文无关文法的描述	12
2 · 1 · 2 语义描述	13
2 · 1 · 3 例子	14
2 · 1 · 4 元语言的结合规则	14
2 · 2 词汇	14
2 · 3 空格的使用	15
2 · 4 注释	16

2 · 5	格式控制字符	16
2 · 6	编译程序命令	17
2 · 7	名字及其定义型出现	18
3	模式和类别	22
3 · 1	概述	22
3 · 1 · 1	模式	22
3 · 1 · 2	类别	22
3 · 1 · 3	模式和类别的性质以及它们之间的关系	23
3 · 2	模式定义	24
3 · 2 · 1	概述	24
3 · 2 · 2	同义模式定义	25
3 · 2 · 3	新模式定义	27
3 · 3	模式分类	27
3 · 4	离散模式	29
3 · 4 · 1	概述	29
3 · 4 · 2	整数模式	30
3 · 4 · 3	布尔模式	30
3 · 4 · 4	字符模式	31
3 · 4 · 5	集合模式	31
3 · 4 · 6	范围模式	33
3 · 5	幂集模式	35
3 · 6	引用模式	36
3 · 6 · 1	概述	36
3 · 6 · 2	受限引用模式	36
3 · 6 · 3	自由引用模式	37

3·6·4 行模式	37
3·7 过程模式	38
3·8 实例模式	40
3·9 同步模式	40
3·9·1 概述	40
3·9·2 事件模式	41
3·9·3 缓冲区模式	41
3·10 输入输出模式	42
3·10·1 概述	42
3·10·2 结合模式	43
3·10·3 访问模式	43
3·11 综合模式	45
3·11·1 概述	45
3·11·2 串模式	45
3·11·3 数组模式	47
3·11·4 结构模式	49
3·11·5 分层结构表示法	56
3·11·6 数组模式和结构模式的布局说明	59
3·12 动态模式	64
3·12·1 概述	64
3·12·2 动态串模式	65
3·12·3 动态数组模式	65
3·12·4 动态参数化结构模式	66
4·单元及其访问	67
4·1 说明	67

4·1·1	概述	67
4·1·2	单元说明	67
4·1·3	单元等同说明	69
4·1·4	基址说明	70
4·2	单元	71
4·2·1	概述	71
4·2·2	访问名字	72
4·2·3	间接引用的受限引用	74
4·2·4	间接引用的自由引用	75
4·2·5	间接引用行	75
4·2·6	串元素	76
4·2·7	串片	77
4·2·8	数组元素	78
4·2·9	数组片	79
4·2·10	结构域	81
4·2·11	单元过程调用	82
4·2·12	单元内部子程序调用	82
4·2·13	单元转换	83
5·值及其操作		84
5·1	同义词定义	84
5·2	原值	85
5·2·1	概述	85
5·2·2	单元内容	86
5·2·3	值名字	86
5·2·4	字面值	88

5·2·4·1	概述	88
5·2·4·2	整数字面值	88
5·2·4·3	布尔字面值	90
5·2·4·4	集合字面值	90
5·2·4·5	空字面值	91
5·2·4·6	字符串字面值	91
5·2·4·7	位串字面值	93
5·2·5	多元组	94
5·2·6	值串元素	101
5·2·7	值串片	102
5·2·8	值数组元素	103
5·2·9	值数组片	104
5·2·10	值结构域	106
5·2·11	表达式转换	106
5·2·12	值过程调用	107
5·2·13	值内部子程序调用	108
5·2·14	启动表达式	114
5·2·15	零元操作符	115
5·2·16	括号表达式	115
5·3	值和表达式	116
5·3·1	概述	116
5·3·2	表达式	117
5·3·3	操作数—1	118
5·3·4	操作数—2	120
5·3·5	操作数—3	122

5.3.6	操作数—4	124
5.3.7	操作数—5	126
5.3.8	操作数—6	128
6	动作	130
6.1	概述	130
6.2	赋值动作	131
6.3	IF动作	133
6.4	CASE动作	135
6.5	DO动作	137
6.5.1	概述	137
6.5.2	FOR控制	138
6.5.3	WHILE控制	144
6.5.4	WITH部分	144
6.6	出口动作	146
6.7	调用动作	146
6.8	结果和返回动作	150
6.9	GO TO动作	152
6.10	断言动作	152
6.11	空动作	152
6.12	CAUSE动作	153
6.13	启动动作	153
6.14	停动作	154
6.15	继续动作	154
6.16	延迟动作	154
6.17	延迟选择动作	155

· 18 发送动作	156
6·18·1 概述	156
6·18·2 发送信号动作	157
6·18·3 发送缓冲区动作	158
· 19 接收选择动作	159
6·19·1 概述	159
6·19·2 接收信号选择动作	159
6·19·3 接收缓冲区选择动作	161
7 · 输入与输出	165
· 1 I/O参考模型	165
7 · 2 结合值	167
7·2·1 概述	167
7·2·2 结合值的属性	167
7 · 3 访问值	168
7·3·1 概述	168
7·3·2 访问值的属性	168
7 · 4 对于输入输出的内部子程序	169
7·4·1 概述	169
7·4·2 结合外界对象	170
7·4·3 分离外界对象	171
7·4·4 访问结合属性	171
7·4·5 修改结合属性	172
7·4·6 连接访问单元	173
7·4·7 切断访问单元	177
7·4·8 访问单元的访问属性	178

7.4.9	数据传送操作	179
8	程序结构	183
8.1	概述	183
8.2	范围和嵌套	185
8.3	Begin-end 分程序	189
8.4	过程定义	189
8.5	进程定义	195
8.6	模块	196
8.7	区域	197
8.8	程序	198
8.9	存储器分配和寿命	198
8.10	分段程序设计结构	199
8.10.1	远端段	199
8.10.2	说明模块、说明区域及上下文组	202
8.10.3	准语句	203
8.10.4	准定义型出现和定义型出现之间的机构	206
9	并发机构	207
9.1	进程及其定义	207
9.2	互斥和区域	208
9.2.1	概述	208
9.2.2	区域性	209
9.3	进程的延迟	212
9.4	进程的再激活	213
9.5	信号定义语句	215
10	总语义性质	217

10.1 模式检查	217
10.1.1 模式和类别的性质	217
10.1.1.1 只读性质	217
10.1.1.2 可参数化模式	217
10.1.1.3 引用性质	217
10.1.1.4 被标记的参数化性质	218
10.1.1.5 非值性质	218
10.1.1.6 根模式	219
10.1.1.7 得结果类别	219
10.1.2 模式和类别上的关系	219
10.1.2.1 概述	219
10.1.2.2 模式上的等价关系	220
10.1.2.3 读相容关系	229
10.1.2.4 动态读相容关系	230
10.1.2.5 可限制关系	231
10.1.2.6 模式和类别之间的相容性	232
10.1.2.7 类别之间的相容性	233
10.1.3 分情况选择	234
10.1.4 语义范畴的定义和摘要	237
10.1.4.1 名字	238
10.1.4.2 单元	240
10.1.4.3 表达式和值	240
10.1.4.4 内部子程序调用	241
10.1.4.5 其它语义范畴	242
10.2 可见性和名字的约束	242

10·2·1	可见性的分级	242
10·2·2	可见性条件和名字的约束	243
10·2·3	隐含名字串	244
10·2·4	范围中的可见性	247
10·2·4·1	概述	247
10·2·4·2	可见性语句	248
10·2·4·3	前缀更名子句	248
10·2·4·4	移出语句	251
10·2·4·5	移入语句	254
10·2·5	域名字的可见性	257
11·	异常处理	258
11·1	概述	258
11·2	处理程序	258
11·3	处理程序的识别	259
12·	实现任选	261
12·1	实现定义的内部子程序	261
12·2	实现定义的整数模式	262
12·3	实现定义的寄存器名字	262
12·4	实现定义的进程名字和异常名字	262
12·5	实现定义的处理程序	263
12·6	实现定义的可引用性	263
12·7	语法任选	263
附录A	CHILL字符集	
附录B	特殊符号和组合字符	
附录C	特殊的简单名字串	

~10~

附录 D 程序例子

附录 E 语法汇集

1 绪 论

本建议书定义了 CCITT 高级程序设计语言 CHILL。CHILL 代表 CCITT High Level Language (CCITT 高级语言)。

CHILL 的另一种定义, 即严格数学形式的定义“CHILL 的形式定义”, 被包括在 CCITT 手册中。另一本 CCITT 手册“CHILL 引论”是作为对该语言的介绍。

本章对 CHILL 的功能进行了非形式的描述。该语言的定义在第二章中叙述。

1.1 概 述

CHILL 最初是为存贮程序控制 (SPC) 电话交换的程序设计而设计的语言。然而, 对于其它的应用 (如: 报文交换、分组交换、模拟等) 也认为是足够通用的。

CHILL 的设计注意到了下列要求 (参阅 Question 8/XI of the study period 1977~1980):

- 以容许广泛的编译时检查来提高可靠性;
- 能产生高效的目标代码;
- 应该是灵活的和强有力的, 以满足应用范围的要求和对各种硬件的利用;
- 有利于并能促进程序的模块化和结构程序设计;
- 易于学习和使用。

对于在 SPC 电话交换中目前使用的或推荐使用的机器种类, CHILL 程序能以不依赖机器的方式书写。

CHILL不试图对上面提到的每一种应用都提供一种特殊的构造,而宁可提供一般的基本构造,这些基本构造有各种适应特殊应用的能力。

作为语言,CHILL是不依赖机器的。然而,一种特殊的实现可以包括由语言目标所规定的实现。包括这样目标的程序,一般来说是不可移植的。

CHILL的设计考虑到从源程序到目标代码的编译。它没有采用特别的设计,去使得单遍扫描成为可能或使得编译程序体积最小。

为了安全,同时又要避免出现效率上的难以接受的损失,许多检查要能在静态时完成。只有很少一些语言规则仅能在运行时检测。违反这些规则的结果是产生运行时异常。然而,对这些异常的运行时检查是任选的,除非指定了程序员定义的异常处理程序。

1.2 语言的总观

CHILL程序由三个基本部分组成:

- 数据对象的描述;
- 在数据对象上执行的动作的描述;
- 程序结构的描述。

数据对象由数据语句描述(说明语句和定义语句),动作由动作语句描述而程序结构由程序结构语句确定。

CHILL 提供了在给定上下文中数据对象使用上的广泛的静态检查。

在下面各节,给出了各种CHILL概念的摘要。每一节都是对与该节有相同标题的详述该概念的一章的介绍。

1.3 模式和类别

单元附有模式 (*mode*)。单元的模式定义了可以存放在该单元中的值的集合,也定义了与该单元以及它可以包含的值相联系的性质(注意:不是单元的所有性质都由它的模式单独确定)。单元的性质是:大小、内部结构、只读性和可引用性等。值的性质是:内部表示、排序和可应用的操作等。

值附有类别 (*class*)。值的类别确定了可包含值的单元的模式。

CHILL 提供了下列模式分类:

离散模式 (*discrete modes*)

整数模式、字符模式、布尔模式、集合(符号的)模式以及它们的范围模式;

幂集模式 (*powerset modes*)

某些离散模式的元素的集合;

引用模式 (*reference modes*)

受限引用模式、自由引用模式以及用于对单元引用的行模式;

组合模式 (*composite modes*)

串模式、数组模式和结构模式;

过程模式 (*procedure modes*)

可操纵数据对象的过程;

实例模式 (*instance modes*)

对进程的标识;

同步模式 (*synchronisation modes*)

用于进程同步和通信的事件模式和缓冲区模式。

输入、输出模式 (*input-output modes*)

对输入输出操作的结合模式和访问模式。

CHILL 给出了一系列标准模式的名称。能够用模式定义的方法引进程序定义的模式。某些语言构造附有动态模式。动态模式是指其中某些性质只能由动态确定的模式。动态模式总是带有运行时参数的参数化模式。非动态模式被称为静态模式。在 CHILL 程序中显式指明的模式都是静态模式。

在 CHILL 中既没有动态模式也没有类别的表示。它们的引入仅仅用来在元语言 (*meta language*) 中描述静态和动态的上下文条件。

1.4 单元及其访问

单元是 (抽象的) 位置。值可以被存在那里, 也可以从那里获得值。为了存储和获得值, 必须访问单元。

说明语句定义了用作对单元访问的名字。有:

- 1、单元说明;
- 2、单元等同说明;
- 3、基址说明。

第一种说明建立了单元并且确定了对新建单元的访问名字。后两种说明对在其它地方建立的单元, 确定了新的访问名字。

除开单元说明, 新单元也能借助于内部子程序 GETSTACK 或 ALLOCATE 来建立。它们产生一个对新建立的单元的引用值 (见下面)。

单元可以是可引用的。这就意味着对单元存在一个相应的引用值。引用值是作为适用到可引用单元的引用操作的结果来获得。通过对引

用值的间接引用，可以获得被引用的单元。CHILL规定某些单元总是可引用的，同时也留下一些单元，由实现确定其是否是可引用的。单元的可引用性必须是静态确定的。

单元可以是只读的。这意味着对它的访问只能是从其中获得值，而不能向其中存一个新值（除了赋初值）。

单元可以是组合的。这意味着它包含有能被独立访问的子单元。子单元不一定必须是可访问的。若一单元，它至少有一个只读子单元，那么我们说该单元具有只读性。得到子单元（或子值）的访问方法是对串或数组的求子串、求下标和求片（*slicing*）以及对结构的选择。

单元附有模式。若模式是动态的，则该单元被称为动态模式单元。（注意：动态这个字只是对模式而言。尽管运行时单元发生变化，并不意味着它是动态的。单元是动态的，只是说它的性质不能完全静态地确定。）

单元的下列性质，尽管是可静态确定的，但它们不是模式的一部分。

可引用性：单元是否存在引用值；

存贮种类：单元是否被静态分配；

区域性（*regionality*）：单元是否被说明在区域（*region*）内。

1.5 值及其操作

值是能在其上定义特殊操作的基本对象。一个值或者是（CHILL）定义值或者是未定义值（就CHILL意义上说）。在所说明的上下文中，未定义值的使用导致未定义局面（就CHILL意义上说）的出现。