红色語言(REDL)的非形式

说明书

陈涵生等 译校

四机部一九三二研究所

目　　录

1.0 引 言

1.1 文卷的意图

这个文卷的意图是要提供REDL的一个非形式说明。REDL是一个按照国防部的修订“铁人要求”（1977年7月）设计的并以Pascal为基础的语言，名REDL是“红色语言（Red Language）”的缩写，源出于DOD为本文卷确立的彩色代码。

REDL的语法用语法图形式地给出，语言的语义用英语非形式地描述。语义的非形式说明是有意的，因为在合同的这一阶段，还不需要一个完整的语言形式定义（事实上，也没有这种可能）。对这个文卷里的描述打算尽量予以完善，以便读者能够判断REDL对“铁人要求”以及由DOD高级语言工作小组确立的其它鉴定原则的顺应性，为了便于描述，我们不时地使用程序设计例子和与pascal的比较，此外，在描述一个特征时，如果对为什么要提出这一特征作出介释是有用的（例如，“铁人”中的一个特殊要求），我们则间或提供动机的陈述。

1.2 结构描述的表示法

REDL与Pascal一样，利用图来表达语言的词法结构和语法结构．每一个词法图指明一个或几个词法单位的结构，每一个语法图指明一个或几个语法类别的结构。顺着一条路线在图中追踪，读者就可以产生由该图表示的词法单位实例或语法类别实例。

1.2.1 词法图

每一个图对特定的一种词法单位定义一组合法的实例。被定义词法单位的名出现在图左上方的园框①里

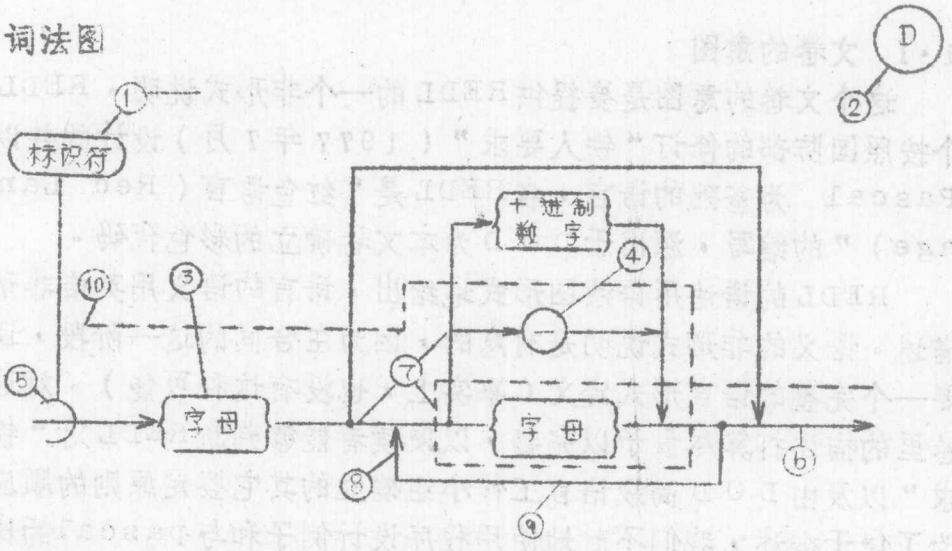
②指示的字母是图的索引，它与特定的图关联，附录C列入了一个关于图索引的交叉引用索引。

二边带花括号的框，如③，表示在附录B中定义的字符分类。

园边框，如④，表示单个字符、邻接的字符序列或词法单位。上

面每一个表示词法单位的圆边框是对该词法单位定义的一个交叉引用。

词法图



要产生词法单位的实例，读者可以从定义框⑤的接合点出发，沿着图自左向右一框一框地走过去，当到达路线的终点⑥便结束。

如果一个图不宜在水平方向前进，就将它分成几个段，段由一行末的三个点和下一行头的三个点来标明。（在上面的词法图中并未示明）。

当读者循图前进到达一个接合点⑦时，可以取从该点发出的任一条路线。

沿着一个收敛路线⑧回溯是不合法的。

有时也可能遇到象⑨那样的潜在的无限循环。

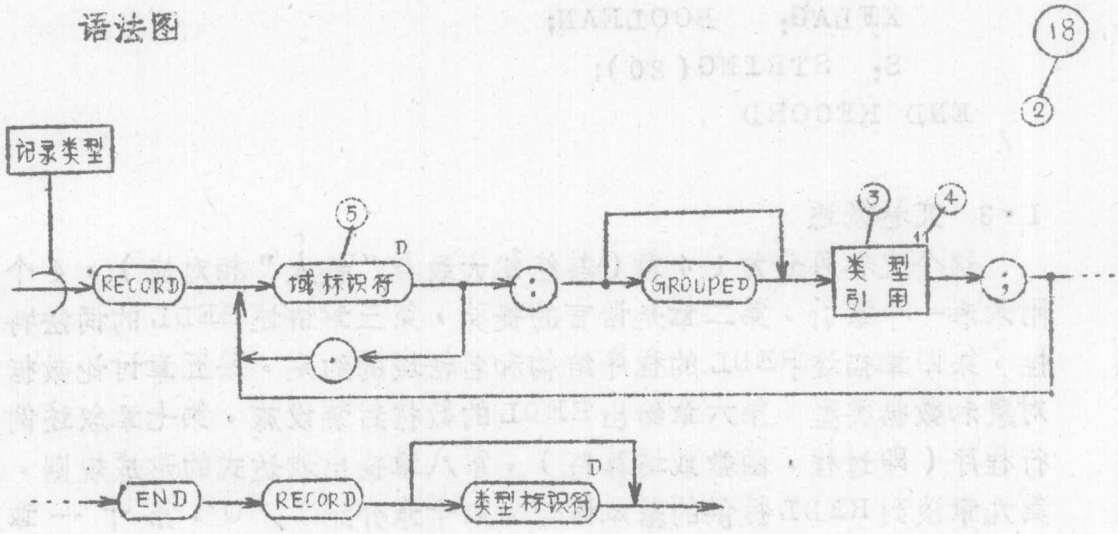
每当遇到一个框时，它的表示字符就右加到沿该路线移动而产生的字符序列中。例如，沿着与点线⑩平行的路线移动便产生“字母—字母”序列（如，A－B或Q－Q）

1·2·2 语法图

语法图的规则与词法图的规则相似，但有如下改变。

被定义语法类别的名在矩形框①中出现，示例定义了《记录类型》语法类别。

语法图



如前节所述，当一个语法类别在正文中引用时，它的名被括在尖括号中。

如②所示的语法图的图索引是一个整型值。

如③那种的矩形框表示的是在其它语法图中定义的语法类别。框④上方的数是与该框有关的语法图索引。

圆边框⑤表示单个字符、邻接的字符序列或称为词法元的词法单位（参见§3·2）。为了便于阐释，词法元“名”和“标识符”经常前缀一个反映它们在语法图的上下文中使用的描述语。在上面所示的语法图中，描述语“域标识符”和“类型标识符”二者指的均是称为“标识符”的词法元。

循着一条路线经过语法图便产生该语法类别的实例。实例是一个词法元序列，序列的任何二个词法元之间可以有一个词法元分隔符（参见§3·3）。如果序列中二个词法元的并置所产生的字符序列会组成一个更长的词法元，则在这二个词法元之间必须放上一个词法元分隔符。如上图所示，在END和RECORD之间必须有词法元分隔符出现。而在域标识符和逗号或分号之间，分隔符可以任选地出现。

语法类别《记录类型》的实例例子是：

```
RECORD   REC_EXAMPLE;
```

```
    XFLAG:   BOOLEAN;
    S:  STRING(20);
END RECORD
```

1·3 文卷概述

这个文卷可分为14章（其组织大致与“铁人”相对应），4个附录和一个索引。第二章是语言的提要，第三章讲述REDL的词法特性。第四章描述REDL的程序结构和名辖域的约定。第五章讨论数据对象和数据类型。第六章给出REDL的数据封装设施。第七章叙述例行程序（即过程，函数或运算符）。第八章提出表达式的形成规则。第九章谈到REDL提供的基本语句。第十章介绍I／O。第十一章涉及例外处理。第十二章陈述多路径和实时设施。第十三章展示依赖机器的特色。第十四章的内容是编译时设施。

附录A列入的一组实际程序设计例子取材于各种嵌入应用。这些程序的意图是既例示REDL的各种特征，又阐明REDL对嵌入系统的应用能力。附录B罗列了与REDL的词法特性有关的信息。附录C是词法图与语法图的字母交叉引用表。附录D包含REDL的“BNF”语法，它与语法图等价，但适宜于在驱动表分析器中直接应用。（从形式上说，REDL语法是LALR(1)的，这意味着使用“由底向上”分析，且可以向前看一个符号而没有回溯）。

1·4 术语

在这个文卷中，词汇“一定（must）”和“可以不（may not）”有时会出现，在英语中这二个词汇含混不清。例如，讲“条件C一定满足”可能意指或者条件C是一个同义重复（这暗示条件C不满足是不可能的）；或者检视条件C的符合是一个经办人的责任（即，如果该条件不满足将是一个错误）。除非由上下文另行指明，我们都认为是后一种介释。（“经办人”就是程序员；除非另作注记，错误将由编译程序查出）。

在词汇“可以不”中也有类似的歧义性，讲“条件C可以不满足”

可能意指或者C不符合是可能的，或者承认C的不符合对经办人是不得已的（否则，就有一个错误产生）。同样，在这个文卷中采用了第二种介释。

2·0 语言提要

本章的材料提供了有关REDL基本特征的总貌

2·1 程序的结构和辖域

一个REDL程序由一列说明（它把名与程序元素相结合）和导令（它对翻译程序提供了补充信息）组成。要执行的动作体现在语句之中，这是通过它们在例行程序（过程、函数和运算符）说明中的制约而在程序里出现。

REDL是一种分程序结构的语言，与pascal 对比之下，在一个辖域里说明的名并非在所有内层辖域里自动已知。如果希望有这种性态，则名必须是渗入的。否则，外部名仅在开辖域和被显式移入的那些辖域里是已知的。

2·2 数据类型

REDL提供了一组丰富多彩的内部类型和类型定义设施。内部类型有BOOLEAN、CHAR、算术类型FIXED 和FLOAT 、字符串和位串类型STRING 与BITS 、依赖机器类型ADDRESS，提供的类型生成器允许定义枚举、数组、记录、机器记录、并、指引元和文件类型。数组、记录和并类型可以参数化，以便书写的例行程序可以随调用的不同而取用不同的变元大小，这样，同一个参数类型的不同对象便有不同的成分个数。内部类型STRING 和BITS都是参数化的。

REDL中类型同一的规则是简单的，每一个类型必须有一个名，类型都是不传导的。名不同的类型应视为不同的类型。

REDL中的每一个类型，无论它是内部的还是用户定义的，都具

有一组由其说明确立的属性。例如，PRECISION就是FLOAT类型的一个属性，同一类型的不同对象，可以有不同的类型属性值。一个数据对象的表示，在该对象创建时确立。这个表示是对象对其类型的每一个属性所具有值的集合。REDL中的赋值和参数传递规则就利用了上述概念。

2·3 数据封装

REDL中的数据封装单位是闭体，这是一个允许用户在内部说明的元素周围指定防护边界的名辖域机制。闭体是一个闭辖域，这意味着为了能在内部引用外面说明的名，它或者必须显式移入的，或者必须是渗入的，如果要在外面引用闭体中说明的元素，则必须将其显式地移出。所以，用户可以有选择地移出那些代表一种意图抽象的闭体元素，也可以隐匿那些视为实现细节的闭体元素。

2·4 例行程序和表达式

过程、函数和运算符均指称为例行程序（例程）。例行程序可以带形式参数。有三种约束分类形式参数：CONST（“输入”），VAR(“引用”)，RESULT(“抄出”)。例行程序可以选用；即同一个名可以与几个例程说明结合，而参数的个数和／或类型决定了对任何调用的相应抉择。

函数和运算符不可以有付作用。REDL对约束分类（唯一允许的是CONST)和从一个函数或运算符内部使用指引元、外部数据、例行程序强加上一些限制。

禁止过程去执行那种会影响程序可靠性和形式语义指明的别名。例如，把经过过程内部其它标识符得到引用的一个数据对象作为VAR参数传递给这个过程是非法的。

REDL提供了一组能由用户通过选用予以扩充的内部运算符。没有关于定义新运算符号或执行别种语法扩展的设施。REDL中表达式的语法把习惯的优先顺序和可结合性规则用于各种运算符。

2·5 语句

REDL有一个基本的语句清单，它罗列了赋值语句，开始语句（对应于分程序结构语言的“分程序”），条件语句，重复语句，控制转移语句和断言语句。选择语句是条件语句的一种，它为辨别一个并对象的“标志”提供了安全而有效的手段。

2·6 输入／输出

REDL为通常的应用级I／O提供了文件生成器和一个库例行程序集合，它还提供了一组定义设施以便在语言内可以定义低级的和／或专用的I／O例行程序。库例行程序基于Pascal，但被定义得使它们用一个明显的“状态”参数返回该例程成功或失败的编码。可定义的设施包括一组编译的类型讯问的特征。

2·7 例行处理

REDL的例外处理设施允许用户提供出错时对程序性态的显式控制。例外可以是预定义的，也可以是用户说明的，一个例外的产生会引起对处理程序的一次搜索；REDL的On 语句提供了一个用户藉以指明处理程序的机制，导令是为了抑制编译程序产生关于预定义例外的检查代码而提供的，它适用于对效率的考虑重于检查的得益那样一类环境。

2·8 多路径和实时设施

REDL中含有一种关于定义并行控制路径的设施（分叉语句），它允许几条路径在以互斥方式存取共享操作（管程闭体）时协同操作，使路径以与时间无关（经过事件）的方式或与时间有关（经过暂停语句）的方式同步执行；语言还包括处理在任何控制路径上失败（X_TERMINATE 例外）的设施。

2·9 依赖机器设施

REDL的依赖机器设施都是密封的。因而，一个想要应用任何这

种设施的程序必须用配置导令使意图明朗化。REDL支撑的机器依赖性包括类型生成器（机器记录）——它能用于定义数据对象的存贮安排。ADDRESS类型、LOC函数、关于排齐或存贮数据的导令、把程序员提供的过程连到异步中断的导令、以及定义汇编语言例程的设施或与用其它语言书写的例程接口的设施。

2·10 编译时设施

REDL的编译时设施包括独立编译的特征（出借导令和存取导令）——它使程序分介而不损失作编译时检查的一般策略成为可能；关于编译时常量说明及其在编译时可计值表达式中使用的设施；关于条件编译和“语义宏”（编译时过程）的设施；关于说明和使用类型“不定”的例程参数的设施。