

附件一

一、基于 C 语法设计的源语言 LittleC

该语言的一个程序由一个块组成，该块中包含可选的声明语句和执行语句。该语言只支持一种数据类型：整型，因而也只有一种类型的变量：整型变量和一种类型的常量：整常数。

该语言支持 +、-、*、/ 四种算术运算，运算式的值为整常数。该语言的选择语句和循环语句中允许使用关系运算表达式来作为控制条件，其值为整常数（非零值表示 true，零值表示 false）。运算符的优先级和结合规则按照 C 语言语法理解。

该语言不支持数组、结构体、指针等复杂数据类型，不支持函数调用，没有 I/O 功能。

1. 文法定义

PROG → BLOCK

BLOCK → { DECLS STMTS }

DECLS → DECLS DECL | empty

DECL → TYPE NAMES ;

TYPE → **int**

NAMES → NAMES , NAME | NAME

NAME → **id**

STMTS → STMTS STMT | empty

STMT → **id** = EXPR ;

STMT → **if** (BOOL) STMT

STMT → **if** (BOOL) STMT **else** STMT

STMT → **while** (BOOL) STMT

STMT → BLOCK

EXPR → EXPR ADD TERM | TERM

ADD → + | -

TERM → TERM MUL UNARY | FACTOR

MUL → * | /

FACTOR → (BOOL) | **id** | **number**

REL→EXPR ROP EXPR

ROP→ > | >= | < | <= | == | !=

2. 词法规则

(1) 名字: 由字母打头后跟字母、数字任意组合的字符串; 长度不超过 15; 不区分大小写; 把下划线看作第 27 个字母;

(2) 常数: 完全由数字组成的字符串; 正数和 0 前面不加符号, 负数在正数前面加-构成; 长度不超过 15;

(3) 关键字、运算符、分隔符仅包含在文法定义中出现过的单词。

(4) 字母表定义为 (1) ~ (3) 中出现的字符的集合; 不在该集合中的符号都以非法字符对待;

二、基于 Pascal 语法设计的源语言 LittleP

由于 Pascal 语言结构严谨, 层次清晰, 语法与 C 语言接近, 也便于理解, 因此本实验抽取 Pascal 语言的一个子集, 稍加改造, 作为源语言, 姑且命名为 LittleP。一个 LittleP 程序由一系列全局数据声明和一个主程序体组成。所有数据采用静态存储分配, 没有 I/O, 只支持一种基本数据类型: 无符号整数。

该语言支持 +、-、*、/ 四种算术运算, 运算式的值为整常数。该语言的选择语句和循环语句中允许使用关系运算表达式来作为控制条件, 其值为整常数 (非零值表示 true, 零值表示 false)。运算符的优先级和结合规则按照 C 语言语法理解。

该语言不支持数组、结构体、指针等复杂数据类型, 不支持函数调用。

1. 文法定义

<程序>→<程序首部><程序体>.

<程序首部>→ **program**<程序名>;

<程序体>→<变量声明><复合语句>

<变量声明>→ **var**<变量定义列表>|<空>

<变量定义列表>→ <变量定义列表>; <变量定义>|<变量定义>

<变量定义>→<变量名列表>: <类型>

<变量名列表> → <变量名列表>,<变量名>|<变量名>

<类型> → **integer**

<复合语句> → **begin** <语句块> **end**

<语句块> → <语句> | <语句块> ; <语句>

<语句> → <赋值语句> | <条件语句> | <循环语句> | <复合语句> | <空>

<赋值语句> → <左部>:= <右部>

<左部> → <变量名>

<右部> → <算术表达式>

<条件语句> → **if** (<关系表达式>) **then** <语句> **else** <语句>

<循环语句> → **while** (<关系表达式>) **do** <语句>

<关系表达式> → <算术表达式> <关系运算符> <算术表达式>

<算术表达式> → <项> | <算术表达式> <加运算符> <项>

<项> → <因子> | <项> <乘运算符> <因子>

<因子> → <变量名> | (<算术表达式>) | <整数>

<程序名> → <标识符>

<变量名> → <标识符>

<标识符> → <字母> | <标识符> <字母> | <标识符> <数字>

<整数> → <数字> | <整数> <数字>

<关系运算符> → < | <= | > | >= | = | <>

<加运算符> → + | -

<乘运算符> → * | /

<字母> → a|b|...|x|y|z

<数字> → 1|2|3|4|5|6|7|8|9|0

2. 词法规则

(1) 名字：由字母打头后跟字母、数字任意组合的字符串；长度不超过 15；

不区分大小写；

(2) 常数：完全由数字组成的字符串；没有正负号；长度不超过 15；

(3) 关键字、运算符、分隔符仅包含在文法定义中出现过的单词。

(4) 字母表定义为 (1) ~ (3) 中出现的字符的集合；不在该集合中的符号都以非法字符对待；

附件二

对自定义源语言的要求：

- (1) 该语言应该至少包含一种数据类型：整型，能够表达整型的变量和常量；
- (2) 能够进行简单的算术运算；
- (3) 能够表达三种常用的语义结构：顺序、选择和循环；
- (3) 程序有唯一的入口/起点和唯一的出口