

Вариант 66 (***)

Разработать систему для управления клеточным роботом, осуществляющим передвижение по клеточному лабиринту. Клетка лабиринта имеет форму квадрата.

Робот может передвинуться в соседнюю клетку в случае отсутствия в ней препятствия.

1. Разработать формальный язык для описания действий клеточного робота с поддержкой следующих литералов, операторов и предложений:

- Знаковых целочисленных литералов в десятичном формате (INT);
- Логических литералов **true** и **false** (BOOL); логические константы и выражения преобразуются к знаковым целочисленным как 1 и 0 соответственно, целочисленные к логическим 0 – false, все остальное – true;
- Перечислимый тип CELL = {EMPTY, WALL, EXIT, UNDEF} перечислимый тип не преобразуется ни в какие другие типы;
- Блок объявления переменных и констант в соответствующих форматах:
 - Блок объявления переменных может содержать объявление нескольких переменных, в программе может быть несколько таких блоков
 - Возможна инициализация многомерных массивов в блоке VALUES, в этом случае размерность этого блока должна совпадать с объявленной в блоке DIMENSIONS, иначе это семантическая ошибка

<VARDECLARATION> <VAR = имя переменной [CONST = TRUE | FALSE]> <TYPE> INT | BOOL | CELL </TYPE> [<DIMENSIONS count = количество пространств> <DIMENSION> максимальный индекс элемента </DIMENSION> </DIMENSIONS>] [<VALUES> <VALUE> арифметическое или логическое выражение </VALUE> </VALUE>]</VAR> <VAR ...>...</VAR>... </VARDECLARATION>;

Применяется строгая типизация, если преобразование не определено и типы не совпадают, то это семантическая ошибка.

- Обращение к переменной
 - **<VAR name = <имя переменной> /VAR>**
 - **<VAR name = <имя переменной> <DIM> <INDEX> арифметическое выражение </INDEX> </DIM> </VAR>**
- Операторов присваивания;
 - Присваивание может быть на уровне элементов массива, либо на уровне всего массива (переменной), в этом случае размерность переменных и выражений должна совпадать.
 - VALUE присваивается все переменным объявленным в блоке TO
- **<ASSIGN> <VALUE> выражение </VALUE> <TO> <обращение к переменной 1> [<обращение к переменной 2> ...]</TO> </ASSIGN>**
- Арифметических операторов бинарных SUB, DIV и мультиарных ADD, MUL (результат временная переменная), для операторов ADD и MUL может применяться сокращенная форма записи (они являются мультиарными):
 - **<ADD|MUL> выражение 1 [выражение 2 выражение 3 ...]</ADD|MUL>**
 - **<SUB|DIV> выражение 1 выражение 2</SUB|DIV>**
- Унарного логического оператора (результат временная переменная)
 - **<NOT> <логическое выражение | вызов процедуры> </NOT>;**
- Мультиарных логических операторов OR и AND операторов (результат временная переменная):
 - **<OR|AND> выражение 1 [выражение 2 выражение 3 ...]</OR|AND>**
- Операторов поиска максимума и минимума (результат временная переменная):
 - **<MAX|MIN> выражение 1 [выражение 2 выражение 3 ...]</MAX|MIN>**
- Операторов проверки на равенство (результат временная переменная):
 - **<EQ> выражение 1 [выражение 2 выражение 3 ...]</EQ>**

- Операторов цикла
 - **<WHILE> <CHECK> <выражение> </CHECK> <DO> <предложение языка 1> [<предложение языка 2> ... группа предложений]</WHILE>**
- Условных операторов **<SWITCH> <CONDITION> <CHECK> <выражение> </CHECK> <DO> <предложение языка 1> [<предложение языка 2> ... группа предложений]</DO> </CONDITION> [<CONDITION>...</CONDITION>...]</SWITCH>;**
- Операторов управления роботом
 - Оператор перемещения роботом LEFT, RIGHT, UP, DOWN, если робот сталкивается со стеной, то он ломается и дальнейшее выполнение программы становится не возможно.
 - **<LEFT|RIGHT|UP|DOWN> выражение </ LEFT|RIGHT|UP|DOWN >**
 - Осмотр окрестностей с помощью спутниковых роботов **<SENDDRONS>выражение</SENDDRONS>**, робот выпускает в случайных направлениях роботов-разведчиков, которые передвигаются по случайной траектории (без возврата назад); ресурс передвижения разведчика не более пяти клеток; количество роботов разведчиков ограничено некоторой величиной; оператор возвращает временную переменную содержащая матрицу соответствующей размерности, содержащей состояние разведанных спутниками.
 - Запись в переменную оставшегося количества роботов-разведчиков **<GETDRONSCOUNT>** имя переменной **</GETDRONSCOUNT>**
- Описатель функции
 - **<FUNC name = имя функции | main> предложения языка </FUNC>**. Возврат значений из функции и передача значений в функции происходит через глобальные переменные. Функция является отдельной областью видимости. Функция не может быть объявлена внутри другой функции. Точкой входа в программу является функция с именем main.
- Оператор вызова процедуры
 - **<CALL>имя функции </CALL>**
- Описатель программы состоит из функций
 - **<PROGRAM> <описатель функции | блок объявления переменных> [<описатель функции | блок объявления переменных> ...]</PROGRAM>**
В корректной программе обязательно должна быть функция с именем main

Язык является регистрозависимым.

2. Разработать с помощью flex и bison интерпретатор разработанного языка. При работе интерпретатора следует обеспечить контроль корректности применения языковых конструкций (например, инкремент/декремент константы); грамматика языка должна быть по возможности однозначной.
3. На разработанном формальном языке написать программу для поиска роботом выхода из лабиринта. Описание лабиринта и начальное положение робота задается в текстовом файле.