|  |  |
| --- | --- |
| Описание: для прик эмбл | |
| МИНОБРНАУКИ РОССИИ | |
| Федеральное государственное бюджетное образовательное учреждение  высшего профессионального образования  **«Московский технологический университет»** | |
| Институт «Комплексной безопасности и специального приборостроения» | |
| Кафедра «Управление и моделирование систем**»**(КБ-3) | |

|  |  |
| --- | --- |
| **ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №1** | |
| **по дисциплине** | |
| **«Дискретная Математика»** | |
|  | |
| Выполнил студент группы БКБО-02-16 | Кулагин Юрий  Вячеславович |
| Принял | Русаков А. М. |

|  |  |  |
| --- | --- | --- |
| Работы выполнены | «\_\_»\_\_\_\_\_\_\_2017 г. | *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_2017 г. | *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* |

*Москва 2017*

**Практическое занятие №1. Разработка синтаксических анализаторов для регулярных языков.**

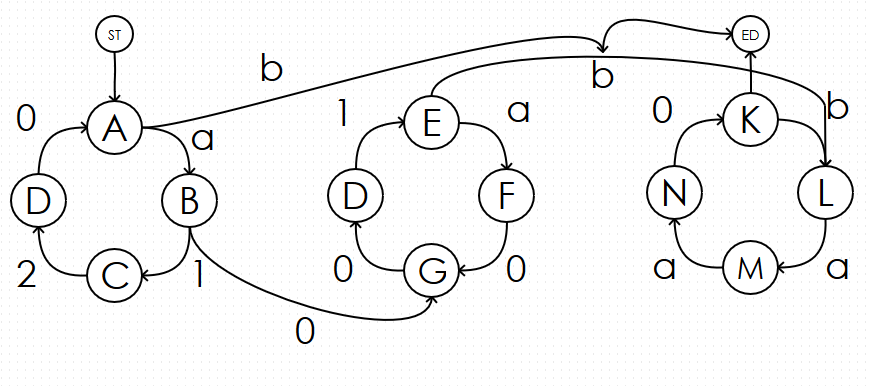
*Цель работы:*

Написание, отладка и проверка работоспособности синтаксического анализатора на основе графа детерминированного конечного автомата, соответствующего заданному регулярному выражению, порождающему конкретный язык.

*Исходные данные варианта задания:*

(a120)\*(a001)\*(baa0)

*Алгоритм решения задачи:*



|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E | F | G | H | K | L | M | N |
| 0 | **B** |  |  |  | **F** |  |  |  |  | **M** | **N** |  |
| 1 |  | **C** |  |  |  |  |  | **E** |  |  |  |  |
| 2 |  |  | **D** |  |  |  |  |  |  |  |  |  |
| a |  | **G** |  | **A** |  | **G** | **H** |  |  |  |  | **K** |
| b | **L** |  |  |  | **L** |  |  |  | **L** |  |  |  |

* Листинг программы:*

*Компилятор: FASM 1.71.60 (flat assembler)*

*Программа была полностью написана с нуля и здесь ужата для лучшего размещения*

*полный листинг со всеми моими комментариями и рабочий образец будет некоторое*

*время доступен по ссылке. (Душу вложил в программу и дьяволу её же продал за такой код,*

*но получил удовлетворение, когда всё заработало как надо)*

*Для запуска рекомендовано использовать DOSBox 0.74, или любой иной эмулятор или*

*реальный 8086 совместимый процессор с настроенной поддержкой прерываний DOS*

|  |  |
| --- | --- |
| org 100h  **JMP** **f\_start**  s\_buff db 254  db 255 dup*(?)*  s\_welcome\_0 db **"Welcome to RegExp checker! By @Jkulvich P.S. it powered by FASM!$"**  s\_welcome\_1 db **"I can check it: (a120)\*(a001)\*(baa0)\*$"**  s\_welcome\_2 db **"Write your string on next line:$"**  s\_checking\_symbols db **"Checking symbols...$"**  s\_incorrect\_symbol db **"Incorrect symbol finded! Must be only: 012ab$"**  s\_correct\_symbols db **"All symbols is correct.$"**  s\_string\_less\_four db **"[NO] - SAD, BUT YOUR STRING CANT BE LESS FOUR$"**  s\_all\_ok db **"[YES] - ALL OK, ITs CONTAIN RegExp!$"**  s\_no\_ok db **"[NO] - SAD, BUT YOUR STRING IS NOT CONTAIN RegExp$"**  **f\_start:**  **MOV** dx*,*s\_welcome\_0  **CALL** **f\_write**  **CALL** **f\_EOL**  **MOV** dx*,*s\_welcome\_1  **CALL** **f\_write**  **CALL** **f\_EOL**  **MOV** dx*,*s\_welcome\_2  **CALL** **f\_write**  **CALL** **f\_EOL**  **CALL** **f\_readline**  **CALL** **f\_EOL**  **CALL** **f\_symbolscheck**  **CALL** **f\_getlength**  **CMP** al*,*4  **JB** **f\_exit\_because\_less**  **CALL** **f\_check**  **CALL** **f\_end**  **f\_check:**  **MOV** dl*,***">"**  **CALL** **f\_print**  **MOV** dh*,*0  **MOV** al*,*dh  **CALL** **f\_getchar**  **CMP** al*,***"a"**  **JZ** **f\_pA**  **CMP** al*,***"b"**  **JZ** **f\_pK**  **CALL** **f\_exit\_because\_nocontain**  **RET**  **f\_pA:**  **MOV** dl*,***"A"**  **CALL** **f\_print**  **CALL** **f\_check\_eol\_exit**  **MOV** al*,*dh  **CALL** **f\_getchar**  **ADD** dh*,*1  **CMP** al*,***"a"**  **JZ** **f\_pB**  **CMP** al*,***"b"**  **JZ** **f\_pL**  **CALL** **f\_exit\_because\_nocontain**  **RET**  **f\_pB:**  **MOV** dl*,***"B"**  **CALL** **f\_print**  **MOV** al*,*dh  **CALL** **f\_getchar**  **ADD** dh*,*1  **CMP** al*,***"1"**  **JZ** **f\_pC**  **CMP** al*,***"0"**  **JZ** **f\_pG**  **CALL** **f\_exit\_because\_nocontain**  **RET**  **f\_pC:**  **MOV** dl*,***"C"**  **CALL** **f\_print**  **MOV** al*,*dh  **CALL** **f\_getchar**  **ADD** dh*,*1  **CMP** al*,***"2"**  **JZ** **f\_pD**  **CALL** **f\_exit\_because\_nocontain**  **RET**  **f\_pD:**  **MOV** dl*,***"D"**  **CALL** **f\_print**  **MOV** al*,*dh  **CALL** **f\_getchar**  **ADD** dh*,*1  **CMP** al*,***"0"**  **JZ** **f\_pA**  **CALL** **f\_exit\_because\_nocontain**  **RET**  **f\_pE:**  **MOV** dl*,***"E"**  **CALL** **f\_print**  **CALL** **f\_check\_eol\_exit**  **MOV** al*,*dh  **CALL** **f\_getchar**  **ADD** dh*,*1  **CMP** al*,***"a"**  **JZ** **f\_pF**  **CMP** al*,***"b"**  **JZ** **f\_pL**  **CALL** **f\_exit\_because\_nocontain**  **RET**  **f\_pF:**  **MOV** dl*,***"F"**  **CALL** **f\_print**  **MOV** al*,*dh  **CALL** **f\_getchar**  **ADD** dh*,*1  **CMP** al*,***"0"**  **JZ** **f\_pG**  **CALL** **f\_exit\_because\_nocontain**  **RET**  **f\_pG:**  **MOV** dl*,***"G"**  **CALL** **f\_print**  **MOV** al*,*dh  **CALL** **f\_getchar**  **ADD** dh*,*1  **CMP** al*,***"0"**  **JZ** **f\_pH**  **CALL** **f\_exit\_because\_nocontain**  **RET**  **f\_pH:**  **MOV** dl*,***"H"**  **CALL** **f\_print**  **MOV** al*,*dh  **CALL** **f\_getchar**  **ADD** dh*,*1  **CMP** al*,***"1"**  **JZ** **f\_pE**  **CALL** **f\_exit\_because\_nocontain**  **RET**  **f\_pK:**  **MOV** dl*,***"K"**  **CALL** **f\_print**  **CALL** **f\_check\_eol\_exit**  **MOV** al*,*dh  **CALL** **f\_getchar**  **ADD** dh*,*1  **CMP** al*,***"b"**  **JZ** **f\_pL**  **CALL** **f\_exit\_because\_nocontain** | **RET**  **f\_pL:**  **MOV** dl*,***"L"**  **CALL** **f\_print**  **MOV** al*,*dh  **CALL** **f\_getchar**  **ADD** dh*,*1  **CMP** al*,***"a"**  **JZ** **f\_pM**  **CALL** **f\_exit\_because\_nocontain**  **RET**  **f\_pM:**  **MOV** dl*,***"M"**  **CALL** **f\_print**  **MOV** al*,*dh  **CALL** **f\_getchar**  **ADD** dh*,*1  **CMP** al*,***"a"**  **JZ** **f\_pN**  **CALL** **f\_exit\_because\_nocontain**  **RET**  **f\_pN:**  **MOV** dl*,***"N"**  **CALL** **f\_print**  **MOV** al*,*dh  **CALL** **f\_getchar**  **ADD** dh*,*1  **CMP** al*,***"0"**  **JZ** **f\_pK**  **CALL** **f\_exit\_because\_nocontain**  **RET**  **f\_symbolscheck:**  **MOV** dx*,*s\_checking\_symbols  **CALL** **f\_write**  **CALL** **f\_EOL**  **CALL** **f\_getlength**  **MOV** ah*,*0h  **MOV** cx*,*ax  **f\_symbolscheck\_cycle:**  **MOV** ax*,*cx  **SUB** al*,*1  **CALL** **f\_getchar**  **CMP** al*,***"0"**  **JZ** **f\_symbolscheck\_eoc**  **CMP** al*,***"1"**  **JZ** **f\_symbolscheck\_eoc**  **CMP** al*,***"2"**  **JZ** **f\_symbolscheck\_eoc**  **CMP** al*,***"a"**  **JZ** **f\_symbolscheck\_eoc**  **CMP** al*,***"b"**  **JZ** **f\_symbolscheck\_eoc**  **MOV** dx*,*s\_incorrect\_symbol  **CALL** **f\_write**  **CALL** **f\_end**  **f\_symbolscheck\_eoc:**  **LOOP** **f\_symbolscheck\_cycle**  **MOV** dx*,*s\_correct\_symbols  **CALL** **f\_write**  **CALL** **f\_EOL**  **RET**  **f\_getchar:**  **MOV** ah*,*0  **ADD** ax*,*2  **ADD** ax*,*s\_buff  **MOV** si*,*ax  **LODSB**  **RET**  **f\_getlength:**  **MOV** ax*,*1  **ADD** ax*,*s\_buff  **MOV** si*,*ax  **LODSB**  **RET**  **f\_readline:**  **MOV** ah*,*0Ah  **MOV** dx*,*s\_buff  **INT** 21h  **RET**  **f\_print:**  **MOV** ah*,*06h  **INT** 21h  **RET**  **f\_write:**  **MOV** ah*,*09h  **INT** 21h  **RET**  **f\_EOL:**  **MOV** dl*,*0Ah  **CALL** **f\_print**  **RET**  **f\_exit\_because\_less:**  **MOV** dx*,*s\_string\_less\_four  **CALL** **f\_write**  **CALL** **f\_end**  **RET**  **f\_exit\_because\_nocontain:**  **CALL** **f\_EOL**  **MOV** dx*,*s\_no\_ok  **CALL** **f\_write**  **CALL** **f\_end**  **RET**  **f\_exit\_because\_ok:**  **CALL** **f\_EOL**  **MOV** dx*,*s\_all\_ok  **CALL** **f\_write**  **CALL** **f\_end**  **RET**  **f\_check\_eol\_exit:**  **CALL** **f\_getlength**  **CMP** dh*,*al  **JZ** **f\_exit\_because\_ok**  **RET**  **f\_end:**  **MOV** ah*,*4Ch  **INT** 21h |

*Трассировка:*

|  |  |
| --- | --- |
| a120 | + |
| a120a120 | + |
| a120a001baa0 | + |
| a001baa0 | + |
| a | - |
| a120a001a120 | - |
| a001baa | - |
| baa0baa0 | + |
| baa0baa | - |
| baa0baa0ba | - |
| a001a001 | + |
| a001a001baa0 | + |
| a001a001a120 | - |
| dfhsjfb | - |
| a121 | - |