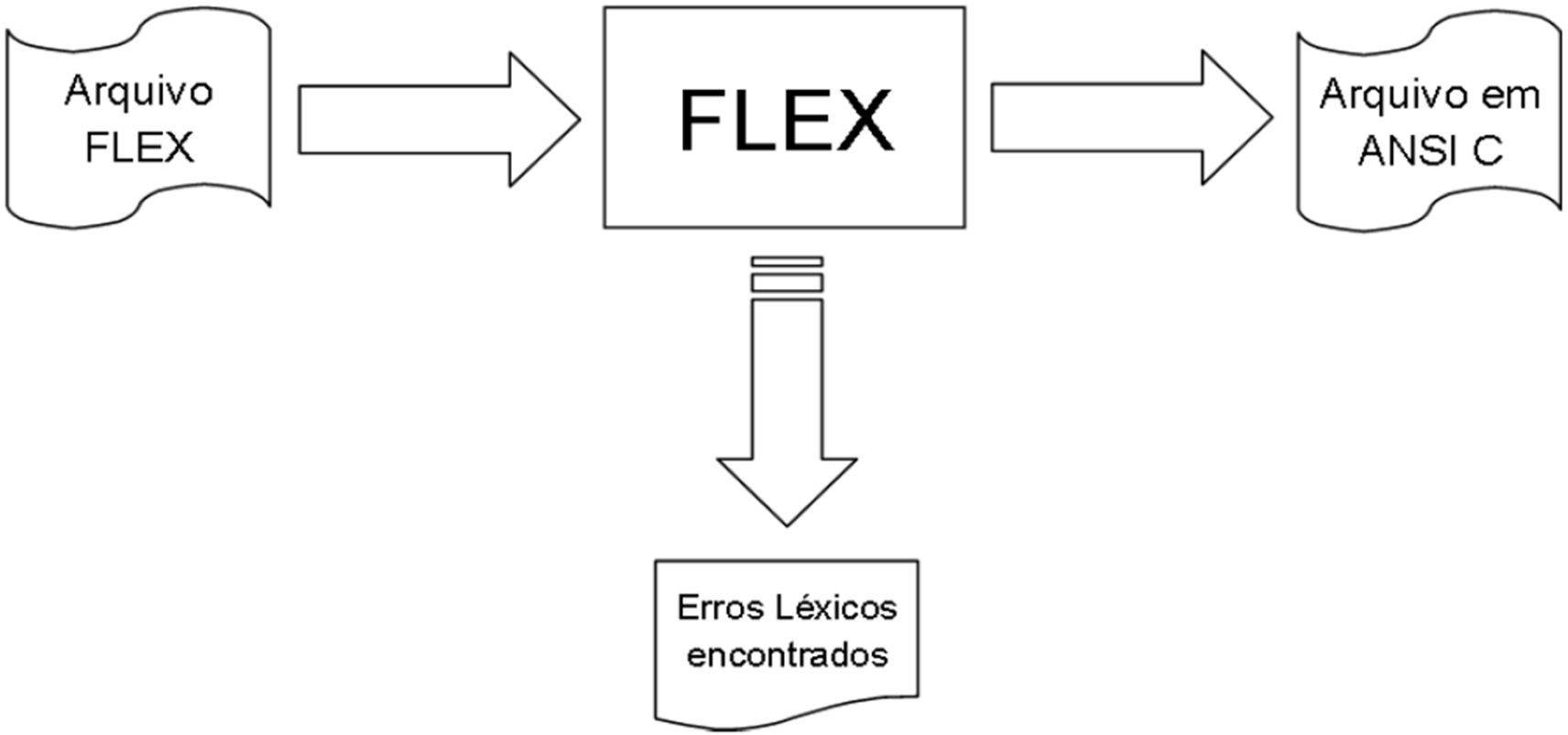
Flex

Flex: Introdução

- O **Flex** (1987) é uma ferramenta para a construção de programas que gerenciam estruturas de entrada.
- O Flex é a evolução do programa **Lex** (1975)**.**
- Foi originalmente desenvolvido para a construção de compiladores, sendo utilizado na geração de **analisadores léxicos**.
- O Flex recebe como entrada, basicamente, uma sequência de expressões regulares; e o que fazer quando um padrão é encontrado (ações).

Flex: Geração do Léxico



Flex: Estrutura do Arquivo de Entrada

Os arquivos de entrada possuem, em geral, a extensão **`.l`** e são constituídos de 3 seções, delimitadas pelos caracteres **%%**.

Definições [opcional]

%%

Regras {ação} [padrão]

%%

Código Auxiliar em C/C++ [opcional]

Flex: Estrutura do Arquivo de Entrada

- A seção "***Definições***" é utilizada para a definição de macros e porções de código em C/C++. Toda porção de código C/C++ deve estar delimitado por ***%{*** e ***%}***.

- As "***Regras***" podem ser entendidas como a funcionalidade principal do analisador léxico por conter as estruturas e regras da análise. Cada regra deve conter uma sequência válida para cada estrutura encontrada, incluindo todas as palavras reservadas diferenciando caracteres maiúsculos de minúsculos, caracteres de tabulação, nova linha e espaços em branco. Todos eles são lidos de forma literal e armazenados na variável "***yytext***" do tipo <*char>, criada pelo programa FLEX.

- A seção "***Código Auxiliar em C/C++***" pode ser utilizada opcionalmente para descrever rotinas auxiliares, porém, no caso do programa escrito para o FLEX ser independente, é necessário que essa seção contenha a função ***main()***.

Flex: Expressões Regulares

- Cada caractere "não especial" representa a si mesmo.
  Ex.: **"a"** corresponde ao caractere **a**
  "if" corresponde à *string* **if**

- **.** (ponto): Representa qualquer caractere (só 1), exceto o fim-de-linha (EOL).

- **[c]** Conjuntos de caracteres: representam 1 só caractere.
  Ex.: **[a-zA-Z]** corresponde a exatamente um único caractere que pode ser qualquer letra do alfabeto de a até z, maiúscula ou minúscula.
- **c*** O caractere **c** repetido zero ou mais vezes.
- **c+** O caractere **c** repetido uma ou mais vezes.
- **c?** "Opcional" – o caractere **c** zero ou uma vez.

Flex: Expressões Regulares

- **r|s** "ou", para indicar a que as cadeias são geradas por uma das duas ER's.
  Ex.: **"a" | "b"**

- **(r)** Agrupamento de expressões regulares. Permite, por exemplo fazer com que uma repetição se aplique a mais do que uma ER.
  Ex.: **("a" | "b")***

Flex: Exemplo

```
/* Nao usar a biblioteca do flex*/
%option noyywrap
%{
int chars = 0;
int words = 0;
int lines = 1;
%}

%%

[a-zA-Z]+ { words++; chars+= strlen(yytext); }
\n        { chars++; lines++; }
.         { chars++; }

%%
int main(int argc, char** argv)
{
    yylex();
    printf("%8d%8d%8d\n", lines, words, chars);
    return 0;
}
```

Flex: Geração do Léxico

```
$ flex arquivo.l
```

Com o comando acima, o flex irá gerar um arquivo chamado **`lex.yy.c`**, o qual contém o código fonte do analisador léxico, o qual pode ser compilado:

```
$ gcc lex.yy.c -o programa
```

O programa gerado já pode ser executado:

```
$ ./programa < arquivo_de_entrada
```