

Compiladores

Exercício 2

Escreva, em C ou em C++, um *parser* $LL(1)$ para a seguinte gramática (os símbolos terminais/tokens são $i, =, e, o, n, (,)$):

$$\begin{aligned} A &\rightarrow i = E \\ E &\rightarrow E e D \mid D \\ D &\rightarrow D o L \mid L \\ L &\rightarrow i \mid n i \mid (E) \mid n(E) \end{aligned}$$

Obs: na prática, a gramática descreve atribuições de expressões booleanas, considerando i como identificadores; e como operador de conjunção/e; o como operador de disjunção/ou; e n como operador de negação/não.

Exemplo de entrada	Exemplo de saída
<code>i=(io(ieni))</code>	entrada valida

Exemplo de entrada	Exemplo de saída
<code>i=(i(ieni))</code>	erro sintatico

Exemplo de entrada	Exemplo de saída
<code>i=ioien(ioi)</code>	entrada valida

- Note que a gramática dada *não* é viável para um *parser* $LL(1)$. Assim, primeiramente transforme a gramática dada em uma gramática $LL(1)$;
- Além do programa em si, envie também um arquivo (PDF, JPG, ou PNG, feito no computador ou foto de feito à mão) mostrando a gramática transformada e a tabela de predição $LL(1)$.

Orientações

- Envie um arquivo .c ou .cpp com sua solução, além de um arquivo de imagem conforme descrito acima;
- Siga o formato de saída apresentado fielmente, para facilitar a correção;
- Não são aceitos envios fora do prazo;
- O envio do exercício é individual, mas ele pode ser feito em grupo.