

DESAFIO – PARSER LL(1)

Para que seu trabalho seja avaliado você deverá postar, no ambiente virtual de aprendizagem, na área reservada para este fim, dois links com a solução do seu trabalho. O primeiro destes links deve apontar para arquivos no ambiente repl.it, onde será possível executar sua solução. O segundo link deve apontar para arquivos contendo os mesmos códigos, contudo hospedado no ambiente Github. **Links que apontem para códigos diferentes em ambientes diferentes serão provocarão o zeramento do trabalho.**

Todos os códigos enviados devem conter o enunciado que está sendo resolvido, na forma de comentário, em pelo menos um dos arquivos enviados. Todos os arquivos de código devem **conter, na primeira linha, em forma de comentário, o nome completo do aluno.**

Você deve ler todo este documento antes de começar e considerar o seguinte código de ética:

- I. Você poderá discutir todas as questões com seus colegas de classe, professores e amigos. Poderá também consultar os livros de referência da disciplina, livros na biblioteca, virtual ou não, e a internet de forma geral e abrangente nos idiomas que desejar. Contudo o trabalho é seu e deverá ser escrito por você. Cópia é plágio.

Para conseguir os pontos referentes a esta avaliação você deverá seguir as instruções apresentadas no item Enunciado deste documento.

OBJETIVO

Pesquisar e praticar. Pesquisar os conteúdos que irão complementar o material apresentado em sala ou nos livros sugeridos na ementa e praticar estes mesmos conceitos. Esta é uma oportunidade para aprimorar sua formação e se destacar profissionalmente.

METODOLOGIA UTILIZADA

Você pode e deve fazer uso das ferramentas disponíveis online para busca de informações. Pode e deve consultar seus colegas de classe e **pode mandar dúvidas para o professor até o dia anterior a data limite de entrega.** Observe as seguintes indicações:

- ao usar o Google, lembre-se de usar os comandos de busca para melhorar a qualidade dos resultados.
- faça sua pesquisa considerando apenas arquivos no formato pdf, disponíveis em instituições de ensino, em qualquer idioma, publicados nos últimos 5 anos;
- Caso seja necessário regidir algum texto para este Trabalho, o texto deverá ser escrito segundo as normas da ABNT. Use apenas as normas que fazem sentido para o trabalho de pesquisa que você está desenvolvendo. Preocupe-se com as fontes, espaçamentos, formato de parágrafos e citações.

Por fim, lembre-se que nenhum trabalho, exercício, ou pesquisa científica, ou acadêmica, admite qualquer tipo de plágio e que todos os conceitos que você trazer para o seu trabalho deverão ser acompanhados da citação correta. Lembre-se também que todos os trabalhos enviados passarão por um sistema de avaliação de plágio e que **trabalhos contendo plágio serão zerados.**

ENUNCIADO

Para obter os pontos relativos a este trabalho, você deverá fazer um programa, usando a linguagem de programação Python, C++, C, ou Javascript que seja capaz de validar expressões de lógica

proposicional escritas em latex e definir se são expressões gramaticalmente corretas. Você validará apenas a forma da expressão (sintaxe).

A entrada será fornecida por um arquivo de textos que será carregado diretamente na linha de comando, com a seguinte formatação:

1. Na primeira linha deste arquivo existe um número inteiro que informa quantas expressões lógicas estão no arquivo.
2. Cada uma das linhas seguintes contém uma expressão lógica que deve ser validada.

A saída do seu programa será no terminal padrão do sistema e constituirá de uma linha de saída para cada expressão lógica de entrada contendo apenas ou a palavra *valida* ou a palavra *inválida* e nada mais.

Gramática:

FORMULA=CONSTANTE|PROPOSICAO|FORMULAUNARIA|FORMULABINARIA.

CONSTANTE="T"|"F".

PROPOSICAO="A"|"B"|"C"|"D"

FORMULAUNARIA=ABREPAREN OPERADORUNARIO FORMULA FECHAPAREN

FORMULABINARIA=ABREPAREN OPERATORBINARIO FORMULA FORMULA FECHAPAREN

ABREPAREN="("

FECHAPAREN=")"

OPERADORUNARIO="\neg"

OPERATORBINARIO="\vee"|" \wedge"|" \rightarrow"|" \leftarrow"

Cada expressão lógica avaliada pode ter qualquer combinação das operações de negação, conjunção, disjunção, implicação e bi-implicação sem limites na combinação, ou aninhamento, de preposições e operações. Os valores lógicos True e False estão representados na gramática e, como tal, podem ser usados em qualquer expressão de entrada.

A validação de cada expressão será realizada por um parser LL(1) que deverá utilizar o algoritmo indicado nas aulas, utilizando, no mínimo uma pilha, um buffer e uma tabela de parser. A tabela de parser deverá ser criada utilizando-se as técnicas conhecidas como *First* e *Follow*.

Para que o parser possa ser validado, a criação da tabela deve ser feita de forma transparente e automática por meio do carregamento da gramática na forma de um arquivo de texto contendo apenas as linhas explicitadas nas regras de produção da gramática que consta neste documento, tratadas com as técnicas *First* e *Follow*. A tabela de parser criada deverá ser impressa no terminal em forma de tabela com a indicação das linhas e colunas que correspondem aos símbolos terminais e não terminais.

O algoritmo LL(1) será comprovado pela impressão no terminal, do estado da pilha, buffer e escolha de regra na forma de uma tabela, mostrando o passo a passo do algoritmo LL(1).

Observe que na gramática proposta os símbolos não terminais estão entre aspas enquanto os símbolos não terminais estão escritos em letras latinas maiúsculas. Todas as informações necessárias para a gramática estão explicitas nas regras de produção da gramática apresentada neste documento. Por fim, o processo de criação do parser será simplificado se cada expressão for primeiro validada

por analisador léxico criado com máquinas de estados finitos, antes da aplicação do parser. A criação de um analisador léxico não é obrigatória.

Para validar seu trabalho, você deve incluir no repl.it, no mínimo três arquivos contendo números diferentes de expressões proposicionais. O professor irá rodar o seu programa com um arquivo de testes extra.

Os alunos que resolverem fazer em Javascript, poderão fazer apenas um site, hospedado no Github, para o teste e todas as saídas podem estar explicitadas em uma página web.

RUBRICAS DE AVALIAÇÃO

Este trabalho tem apenas duas possibilidades de avaliação:

1. A tabela foi corretamente criada, impressa, o parser LL(1) foi corretamente implementado e todas as expressões dos arquivos de teste foram validadas. Neste caso, o aluno fará juz a nota máxima em todas as tarefas da disciplina Construção de Interpretadores do segundo semestre de 2022.
2. Qualquer erro no programa, ou inadequação ao que está especificado neste documento implica na perda de qualquer valor deste trabalho na avaliação da disciplina Construção de Interpretadores do segundo semestre de 2022.

CUIDADOS QUE VOCÊ PRECISA TOMAR:

Este trabalho pode ser entregue até o dia da penúltima aula. Este trabalho não isenta a realização de qualquer um dos trabalhos da disciplina. Se o aluno não fizer os trabalhos da disciplina e não conseguir entregar o desafio será reprovado.

Programas com códigos idênticos serão zerados. e não terão qualquer impacto na nota do aluno

Certifique-se que o seu código pode ser acessado por alguém além de você. Cada ambiente tem uma regra diferente de postar o código de forma que ele seja acessível. Para não ter problemas, depois que terminar e antes de postar os códigos. Acesse os links usando outra identidade no seu navegador *web* ou peça para algum colega acessar seus links. **Programas com código inacessível serão zerados.**