

PARSERS

Parsers são algoritmos usados para percorrer uma determinada estrutura de dados. Na criação de árvores sintáticas, no domínio da interpretação de código fonte, classificamos os parsers em duas grandes famílias: os bottom-up e os top-down.

Esta dominância é diretamente relacionada ao sentido que a estrutura de dados é percorrida e não tem relação com o resultado. Para realmente entender a conversão de um *string* em uma árvore, precisamos, antes de qualquer coisa, entender como os parsers realizam sua tarefa validando estas *strings* frente a uma gramática.

OBJETIVO

Participar os algoritmos de parser, manualmente, para solidificar o conhecimento destas técnicas.

MATERIAL UTILIZADO

Para realizar esta atividade você precisará de um ambiente de desenvolvimento de software, acesso a internet para realização de pesquisas e para a postagem do resultado.

METODOLOGIA DE PESQUISA

Você deverá realizar uma pesquisa na internet para encontrar material para a realização deste trabalho. Neste caso, existem algumas considerações que devem ser feitas:

- Ao usar o Google, lembre 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;
- A sua pesquisa deverá ser escrita segundo as normas da ABNT para a escrita de artigos científicos;
- A parte de conteúdo da sua pesquisa deverá conter entre 2 e 5 páginas.

Lembre-se que uma pesquisa científica não admite qualquer tipo de plágio e que todos os conceitos que você trazer para o seu trabalho deverão ser acompanhados da citação correta.

EXERCÍCIOS PROPOSTOS

1. Construa a árvore de parse para as *strings*: “1-2-3” e “1+2*3” segundo o seguinte conjunto de regras de produção:

$$\begin{aligned} P = \{ & S \rightarrow id; \\ & S \rightarrow num; \\ & S \rightarrow S * S; \\ & S \rightarrow \frac{S}{S}; \\ & S \rightarrow S + S; \\ & S \rightarrow S - S; \\ & S \rightarrow (S) \} \end{aligned}$$

Onde *id* é uma variável e *num*, um número inteiro. Determine se essa gramática é ambígua ou não.

2. Construa a árvore de parse para as *strings*: “1-2-3” e “1+2*3” segundo o seguinte conjunto de regras de produção:

$$\begin{aligned} P = \{ & S \rightarrow S + T; \\ & S \rightarrow S - T; \\ & S \rightarrow T; \\ & T \rightarrow T * F; \\ & T \rightarrow \frac{T}{F}; \\ & F \rightarrow id; \\ & F \rightarrow num; \\ & F \rightarrow S \} \end{aligned}$$

Onde *id* é uma variável e *num*, um número inteiro. Determine se essa gramática é ambígua ou não.

3. Explique, porque existe uma diferença nos resultados e classifique as gramáticas em termos de ambiguidade.

REFERÊNCIAS

AHO, Alfred V.; LAM, Monica S.; SETHI, Ravi; ULLMAN, Jeffrey D. **Compiladores, Princípios, Técnicas e Ferramentas**. 2ª Edição. Rio de Janeiro: Pearson, 2008.

MENEZES, Paulo B. **Linguagens Formais e Autômatos**. 6ª edição. Bookman, 2011

SEBESTA, R. **Conceitos de Linguagens de Programação**. 9ª ed. Editora Bookman, 2011.

TOURING, Alan. **On Computable Numbers, With An Application To The Entscheidungsproblem**, Proceedings of the London mathematical society, 2(1), 230-265.