

João Vitor Couto Oliveira

Linguagens Formais e Teoria da Computação
- Ciência da Computação
Simulador de linguagens regulares

Professor: Celso Olivete Junior

Presidente Prudente

17 de Dezembro de 2021

Resumo

Foi desenvolvido um simulador de linguagens regulares para web, que aceita a criação e testes de expressões regulares, gramáticas regulares e autômatos finitos.

Linguagens e ferramentas utilizadas

Para o desenvolvimento da aplicação foi utilizada a linguagem javascript, a biblioteca RegExp para testar as expressões regulares, o Bootstrap 4.6 para fazer o front-end, a biblioteca VisJs network para implementar os autômatos finitos.

Para editar o código foi utilizado o VisualStudioCode.

Desenvolvimento para Expressões Regulares

Foi criado um campo onde é escrito a expressão e um campo onde pode-se colocar entradas para testar, podendo colocar quantas entradas forem necessárias.

A expressão regular é lida e testada pelo RegExp.

Desenvolvimento para Autômatos Finitos

É aceito autômatos finitos determinísticos (AFD) e autômatos finitos não-determinísticos (AFND) sem caminhos vazios.

Foi utilizado conceitos de grafos para essa implementação, onde a parte visual de desenho e ligação de nós é feita através da biblioteca VisJs Network.

No campo de teste é possível colocar várias entradas.

Quando é criado um nó cria-se também uma array 2d inicialmente com os valores em null, ao ligar os nós com as arestas, é colocado o valor da aresta no array na posição do nó inicial para o nó final.

Quando se coloca uma entrada e clique para testa-lá é rodado uma busca em profundidade no grafo, testando se há algum caminho do nó inicial ao nó final, esse teste é feito pegando o caracter da entrada e comparando com as arestas para verificar se é possível ir para um próximo estado.

Caso consiga chegar a um estado final é retornado true deixando a entrada verde, caso contrário retorna false deixando a entrada vermelha.

Desenvolvimento para Gramáticas Regulares

Temos a aba de testes que o frontend funciona igual aos anteriores, temos um input para definir a letra inicial, há um botão para adicionar o caractere de terminação no último input clicado.

Para adicionar os elementos da gramática há dois inputs separados por uma flecha, onde no lado esquerdo é o nome, aceitando somente letras maiúsculas, e do lado esquerdo as regras, onde se aceita somente gramática linear unitária à direita sendo separada as expressões por |.Clicando no + e adicionada a expressão.

Foi criada uma classe de regras onde se tem o atributo left que representa o nome e right um array com as regras.

Ao clicar para testar é percorrido a entrada verificando se há no elemento Regra inicial há um right que aceite o primeiro caractere da entrada, caso tenha e esse atributo right tenha um elemento não terminal, então é procurado a regra com o atributo left igual a esse elemento. Caso consiga percorrer a entrada completamente e ao chegar no último elemento Regra e não houver um próximo então é retornado true aceitando o teste, caso não consiga percorrer a entrada completamente ou chega no final da entrada e haja somente regras com um próximo elemento é retornado true negando a entrada de teste.