



PROJETO PARA AUTÔMATO FINITO

Cleisson Vieira Raimundi¹

RESUMO: Esse trabalho teve como objetivo a construção de aplicação para geração, eliminação de epsilon transições, determinização e minimização de autômatos finitos. Dando um arquivo .txt com a relação de tokens e/ou GRs de uma linguagem hipotética na entrada a saída é um Autômato Finito Determinístico (AFD), livre de épsilon transições, mínimo sem a aplicação de classes de equivalência entre estados.

¹ Discente do curso de Ciência da Computação, Universidade Federal da Fronteira Sul - *campus* Chapecó

Introdução

Autômatos finitos são reconhecedores de linguagens regulares definidos através de quintuplas da forma $M = (Q, \Sigma, \delta, q_0, F)$. Um autômato finito determinístico (também chamado máquina de estados finita determinística) é uma Máquina de estados finita que aceita ou rejeita cadeias de símbolos gerando um único ramo de computação para cada cadeia de entrada.

Projeto para Autômato Finito

Primeiro foi feita uma carga inicial do autômato, inserindo em uma estrutura de estados a partir de um arquivo contendo apenas os tokens de uma linguagem (tokens.txt). Logo em seguida é feita a leitura da gramática regular da linguagem, que se encontra no mesmo arquivo logo abaixo dos tokens, e inserido os novos estados que foram criados pela gramática na estrutura que já continha o autômato de reconhecimento dos tokens. Isso tudo é feito utilizando tokens.txt e Automatos.py.

Após os passos seguidos acima foi implementado as classes para Eliminar Epsilon Transição, fazer a Determinização, Eliminar Inalcançáveis e então Eliminar os Mortos. Para eliminar os inalcançáveis e mortos foi utilizado busca em profundidade. Tudo isso é feito, principalmente, na Main.py

A estrutura utilizada para a gravação dos estados do autômato foi feita em uma estrutura de dicionário de estados. Cada estado é um dicionário de transições, e cada transição contém uma lista de estados. Foi implementada uma lista de estados para poder inserir transições indeterminísticas. Foi utilizada também a estrutura de conjunto, para representar o alfabeto da linguagem.

Epsilon Transição: também conhecida como lambda, é caracterizada por transições em que é possível passar de um estado para outro sem ler nenhum símbolo de entrada. Quando uma epsilon transição é encontrada deve ser removida para otimizar o processo.

Na implementação para remover essa epsilon transição, por exemplo, é feita da seguinte maneira: do estado X para Y, primeiro substitui em X a transição para Y por todas as transições de Y. Tendo num estado W, uma transição para X por um símbolo z da linguagem, deverá ser adicionado em W uma transição para Y, através do mesmo símbolo z. Se o estado Y for um estado final, então o estado X também será final.

Determinizar: um autômato não determinístico é uma máquina de estados em que para cada par de estado e símbolo pode haver mais de um próximo estado possível, caracterizando uma linguagem computacionalmente irreconhecível.

Para encontrar um determinismo é feito uma sequência básica, cria um novo estado, substitui o indeterminismo pelo novo estado onde aparecer, armazenando as informações dessa substituição para poder fazer o mesmo processo quando surgir o mesmo indeterminismo ao decorrer do processo

Inalcançáveis: existem estados inalcançáveis em um autômato, são estes estados que nunca se pode alcançar partindo do estado inicial. Para remover estes estados foi utilizado busca em profundidade onde percorre todos os estados, passando por todas as transições e adicionando cada estado visitado em uma nova estrutura, parecido com a estrutura inicial do autômato.

Mortos: existem estados mortos em um autômato, são estes estados que jamais alcançam um estado final. Para remover esses estados foi utilizado busca em profundidade onde percorre o autômato partindo de cada um dos estados alcançáveis. Se chegar a um estado onde todas os estados de suas transições já foram visitados, ou chegar à um estado final, o algoritmo mantém o estado no autômato.

Conclusão

Ao dar como entrada ao algoritmo um arquivo contendo os tokens e a gramática do autômato, ele vai realizar os processos de eliminação de epsilon transições, determinização e eliminação de mortos e inalcançáveis, retornando um autômato finito minimizado.

Referências

- Autômato finito determinístico (10h57min de 26 de março de 2019)
https://pt.wikipedia.org/wiki/Aut%C3%B4mato_finito_determin%C3%A9tico
 - Autômato Finito Não Determinístico:
<http://www.decom.ufop.br/anderson/BCC242/AFN.pdf>
 - Python - Dictionary, Tutoriaispoint:
https://www.tutorialspoint.com/python/python_dictionary.htm
 - Como criar minha primeira classe em python, DevMedia:
<https://www.devmedia.com.br/como-criar-minha-primeira-classe-em-python/38912>
 - re — Regular expression operations, Python (Jul 05, 2019):
<https://docs.python.org/3/library/re.html>
- Busca em profundidade (DFS), Paulo Feofiloff, IME-USP (2019-04-08):
https://www.ime.usp.br/~pf/algoritmos_para_grafos/aulas/dfs.html