



Pontifícia Universidade Católica de Minas Gerais

Curso : Ciência da Computação

Disciplina: Fundamentos Teóricos da Computação

PUC Minas Professor : Zenilton Kleber Gonçalves do Patrocínio Júnior

TRABALHO PRÁTICO N.01 (08 PONTOS)

Um autômato finito determinístico (**AFD**) é denominado mínimo se não existir nenhum outro AFD com menos estados que reconheça a mesma linguagem. O objetivo deste trabalho é implementar um programa de obtenha um **AFD** mínimo para determinada linguagem. Para tanto, sua implementação deve receber como entrada um arquivo contendo a descrição de um **AFD** qualquer e gerar como saída outro arquivo contendo a descrição do **AFD** mínimo.

Em 1971, Hopcroft apresenta um método capaz de realizar a minimização de um **AFD** com n estados em $O(n \log n)$. Contudo, em livros e outros materiais didáticos geralmente se apresentam versões com custo $O(n^2)$. Em Blum (1996), apresenta-se uma variação da implementação de custo $O(n^2)$ que juntamente com uma estrutura dados (semelhante à usada por Hopcroft) é capaz de realizar a minimização de um AFD em $O(n \log n)$. Descrições completas, tanto do algoritmo de custo $O(n^2)$ original quanto da proposta de modificação dele, podem ser encontradas no seguinte artigo (disponível também no **Canvas**):

Norbert Blum. 1996. *An $O(n \log n)$ implementation of the standard method for minimizing n -state finite automata*. Information Processing Letters 57, 2 (Jan. 29, 1996), 65–69.
[https://doi.org/10.1016/0020-0190\(95\)00199-9](https://doi.org/10.1016/0020-0190(95)00199-9)

Caso, seja necessário, abaixo é dada uma referência para o artigo de Hopcroft (1971):

Hopcroft, John. *An $n \log n$ algorithm for minimizing states in a finite automaton*. **Theory of machines and computations** (Proc. Internat. Sympos., Technion, Haifa, 1971), New York: Academic Press, pp. 189–196, 1971.

Versão preliminar disponível em: <http://i.stanford.edu/pub/cstr/reports/cs/tr/71/190/CS-TR-71-190.pdf>

Neste trabalho, você deverá implementar os **dois métodos** apresentados por Blum (1996) para gerar um **AFD** mínimo equivalente: (i) o método $O(n^2)$ semelhante àqueles geralmente encontrado em livros didáticos; e (ii) o método modificado com custo $O(n \log n)$. Para tanto, sua(s) implementação(ões) deve(m) receber como entrada um arquivo contendo a descrição de um **AFD** qualquer e gerar como saída outro arquivo contendo o resultado da minimização. Devem ser realizados experimentos que para avaliar o tempo médio gasto para as duas estratégias aplicadas a **AFDs** de diferentes tamanhos.

Além disso, deve-se utilizar como formato para os arquivos de entrada e saída o mesmo padrão adotado pelo simulador **JFLAP versão 7.0** que pode ser encontrado em <http://www.jflap.org/> (favor prestar **muita atenção** na versão a ser utilizada).

Você deverá entregar via **CANVAS**, além dos códigos implementados, um relatório (obrigatoriamente feito em LaTeX) em formato PDF (juntamente com seus códigos-fontes em LaTeX) descrevendo



Pontifícia Universidade Católica de Minas Gerais

Curso : Ciência da Computação

Disciplina: Fundamentos Teóricos da Computação

PUC Minas Professor : Zenilton Kleber Gonçalves do Patrocínio Júnior

detalhes das implementações, dos experimentos e resultados obtidos. O trabalho pode ser desenvolvido e entregue em grupos de até 02 (dois) alunos.

Vale lembrar que o trabalho deve ser desenvolvido e entregue por cada grupo (de forma autônoma e sem cópias) – contudo discussões entre os grupos para melhoria das soluções apresentadas são estimuladas. No caso de trabalhos feitos em grupo, **TODOS os membros devem entregar uma cópia do trabalho via CANVAS.**

Data de Entrega no Canvas : 01/05/2023

OBS: Cópias serão sumariamente zeradas. Caso um dos membros não entregue, mesmo que os outros entreguem, ele ganhará zero. Além disto, a entrega dos fontes em LaTeX é obrigatória (caso contrário o relatório será desconsiderado).