

UNIVERSITY OF COPENHAGEN  
Computer Science Department  
Data-Parallel Compilation  
Lexical analysis & Syntax Tree Construction

William Henrich Due (mcj284)  
Submitted: 5th of April 2024

**Abstract**

Abstract.

## 1 Introduction

Introduction.

## 2 Theory

Hills paper “Parallel lexical analysis and parsing on the AMT distributed array processor” [1] describes a method to obtain the path in a deterministic finite automata given a input string. This section will describe the theory of this method and extend the it for tokenization.

### 2.1 Data-parallel Lexical Analysis

To explain the theory of parallel lexical analysis we first remind the reader of the definition of a deterministic finite automaton.

**Definition 2.1** (DFA). A deterministic finite automata [2] [3] is given by a 5-tuple  $(Q, \Sigma, \delta, q_0, F)$  where.

1.  $Q$  is the set of states where  $|Q| < \infty$ .
2.  $\Sigma$  is the set of symbols where  $|\Sigma| < \infty$ .
3.  $\delta : \Sigma \rightarrow Q \rightarrow Q$  is the transition function.
4.  $q_0 \in Q$  is the initial state.
5.  $F \subseteq Q$  is the set of accepting states.

Note that this definition utilizes currying for the transition function  $\delta$ . This is done in the definition because if we have any two functions  $g = \delta(a)$  and  $f = \delta(a')$  then it follows from composition that for any  $q \in Q$ .

$$g(f(q)) = (g \circ f)(q)$$

---

This allows for an alternative way of determining if a string can be produced by an DFA. Instead of first evaluating  $f(q)$ , then  $g(f(q))$  and then checking if this state is a member of  $F$ . We could instead partially apply  $\delta$  to the symbols and then compose them to a single functions which could be used to determine if a string is valid. This sets the stage for data-parallel lexing, we want to find a way to make the problem into a **map-reduce**.

**Definition 2.2.** A endofunction is a function  $f : A \rightarrow A$  where the domain is equal to its codomain.

## 2.2 Parallel Tokenization

## 3 Conclusion

Conclusion.

## References

- [1] Jonathan M.D Hill. “Parallel lexical analysis and parsing on the AMT distributed array processor”. In: *Parallel Computing* 18.6 (1992), pp. 699–714. ISSN: 0167-8191. DOI: [https://doi.org/10.1016/0167-8191\(92\)90008-U](https://doi.org/10.1016/0167-8191(92)90008-U). URL: <https://www.sciencedirect.com/science/article/pii/016781919290008U>.
- [2] John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation (3rd Edition)*. USA: Addison-Wesley Longman Publishing Co., Inc., 2006. ISBN: 0321455363.
- [3] Wikipedia contributors. *Deterministic finite automaton — Wikipedia, The Free Encyclopedia*. [Online; accessed 4-February-2024]. 2023. URL: [https://en.wikipedia.org/w/index.php?title=Deterministic\\_finite\\_automaton&oldid=1192025610](https://en.wikipedia.org/w/index.php?title=Deterministic_finite_automaton&oldid=1192025610).