On Some Parallel Algorithm for NFA Beterminization[*]

M.E. Abramyan[1,2], N.I. Krainukov[1]

Faculty of Computational Mathematics and Cybernetics, Shenzhen MSU-BIT University, Shenzhen, China,[1]
Algebra and Discrete Mathematics Department, Southern Federal University, Rostov-on-Don, Russian Federation[2]

The many problems are in theory of automata, one from this problem that is the problem of determinization of nondeterministic finite automata (NFA) [1].

The every letter of alphabet Σ produces the transformation of states of DFA A, but for NFA every letter produces the transformation of power set 2^Q of states Q. The subset construction converts an NFA to DFA. Every states of correspondent DFA is the subset of states NFA.

The main idea of algorithm of parallels determinization NFA is like the idea breadth-first search (BFS) in theory algorithms on graphs [2, 3].

For subset construction on parallel determinization NFA, three hash tables are used: **Visited**, **Csstates** (current subsets) and **Nsstates** (next subsets). The Hash table keys are sets of states of the NFA, and Hash table values are unique integers ($key \longrightarrow int$).

For saving a transition table DFA is also used a **TransitionList** of triples (KeyIntCsstates, Letter, KeyIntNsstates). This list consists of already discovered part of transition of the equivalent DFA.

Algorithm parallel determinization NFA is showed in Fig. 1:

```
1: void ParallelDeterminization(Node r)
2: Visited=Csstates=∅; Nsstates=r; /*V,Current, and Next*/
3: r.lev=level=0;
4: for a in Alphabet do
5: repeat
6:  Csstates=Nsstates;
7:   for State s ∈ Csstates do        /*in parallel */
8:    for State n ∈ Transition(s)(a) do /*in parallel */
9:        Add(TransitionList,(s, a, n))
10:        if (n¬∈ Vizited) then
11:           Nsstates = Nsstates ∪ n ; Vizited = Vizited ∪ n
12:        n.lev=level+1
13:   level++
14: until Nsstates==∅
15: od; /* for a in Alphabet */
```

Fig. 1. Algorithm parallel determinization NFA

This paper studied parallelizing determinization algorithm of NFA in two different approaches of HPCGAP and OpenMP experimentally. Our approaches show that implementation the different models for parallelization may produce approximately equal speed if it is executed in the same conditions.

References

1. Lallement G. Semigroups and Combinatorial Applications. NJ, Wiley & Sons, Inc., 1979.
2. Cormen T., Leiserson C. Introduction to Algorithms. MIT Press and McGraw-Hill, 2001.
3. Roosta S.H. Parallel Processing and Parallel Algorithms. Springer, 2000. DOI: 10.1007/978-1-4612-1220-1.

[*] This work is supported by a grant from the research program of Chinese universities "Higher Education Stability Support Program" (Section "Shenzhen 2022 – Science, Technology and Innovation Commission of Shenzhen Municipality") – 深圳市 2022 年高等院校定支持划助目.