Development of A Structural Disciplinary Language for Materials Science

Kou Amano [1], Koichi Sakamoto [1]

[1] *National Institute for Materials Science*

Background: Materials science is based on multiscale and multiphysical disciplines (scientific discipline); therefore, in this field, there are many types of data, models, and terms with various meanings. Although data-driven science have many prospects, it is difficult to operate data on unified discipline (data discipline). However, a well-defined uni-language that treats multimodal forms can help operations. Therefore, we are developing a language that can parse tree or graph structures, enabling the operation of several data formats, models, and dictionaries for materials science. The language is neither a solver nor an analyzer; it is a format converter and a dictionary (syntax) parser, which is connected to solvers and analyzers. The developed language, named “tq”, should satisfy the following: parsing tree structure, parsing graph structure, searching dictionary, matching terms using dictionary, reforming unstructured data to structured data, and conversion to other well-known formats such as JSON. Its challenges are matching or searching tree or graph structures, rewriting of term or phrase (sub-tree) in tree or graph structures based on the similarity (Term Rewriting by Network Similarity (TRNS); pronounced “trans”), daemonizing dictionary system, and parallelizing.

Methods: tq is being developed in C language with POSIX thread and MPI libraries.
The language strucure is defined as following:

```
<T-form>::=<head>|<list>
<head>::=(<reference>|<label><operator><name>)<dim>
<list>::=<head>('('<T-form>(','<T-form>)*')')+
<reference>::='$'<label>|<NULL>
<label>::='#'<number>|<NULL>
<operator>::='$'<string>'$'|<NULL>
<name>::=<string>|<NULL>
<dim>::=('['(<number>(','<number>)*|<NULL>)']')*
<number>:: sequence consisting of '1', '2', '3', '4', '5', '6', '7', '8', '9', '0'
<string>:: sequence consisting of any character except '[', ']', '(', ')', '#', '$', '\n', '\t', ','
<NULL>:: null string
```

Results: Now tq has realized the functions: tree parsing, graph parsing with term reference, JSON output and Wolfram Language output. But currently multi-thread and multi-process function is not implemented.

Conclusions: We have realized a high-performance parser for tree and graph structure, as the language "tq".