INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Design of Web-Based Mathematical Knowledge Repository with LaTeX-like Format OFM

Kohji Ohtsuka[1]
Department of Information Design, Hiroshima Kokusai Gakuin University
6-20-1 Nakano, Aki-ku 739-0321, Japan

Abstract: A repository of Mathematical Knowledge (*MaKR*) is designed as an individual tool, which also works as an assistant that helps thinking about the mathematical subjects. MaKR is a mathematical Web-based system whose contents are glued by an original LaTeX-like formatting language *(low-)OFM* that are designed under the concept of *pushdown automaton*, whose translator form OFM to HTML is written in Perl and named *ofm2html*. For the management of resources, a special-purpose Web browser called *MaKRV* is designed. Main topic in this paper is formatting language *(low-)OFM*.

Keywords: Mathematical knowledge; Web-based repository; LaTeX; pushdown automaton; Perl

1 Introduction

As shown in [1, 8], there are many mathematical Web-based systems, however almost of them are the collection of mathematical publications. Publications are final stage in our academic activity, and our time was devoted mainly to activities that were essentially clerical or mechanical: searching, calculating, plotting, transforming, determining the logical or dynamic consequences of a set of assumptions or hypotheses, preparing the way for a decision or an insight as stated in [5]. Our target is how to construct *repository* of *mathematical knowledge (abbr. MaK)* and we say the collection of publications *archive* for the sake of identification. MaK is a collection of informations in which an individual stores all memos, the lists of references, sketch of proofs, personal notes, retention of facts. In this paper, we write the design of a repository of MaK dubbed *MaKR* which is designed as an individual tool, which also works as an assistant that helps thinking about the subject.

MaKR is a mathematical Web-based system, and work on the platform MS-Windows(XP,Vista). In MaKR, resources are various sorts; text files (HTML, XML, TeX, LaTeX, source codes), PDF-files, MS-Words, MS-Excel, executable binary file, so on. They are glued by an original LaTeX-like formatting language *(low-)OFM* that are designed under the concept of *pushdown automaton*, whose translator form OFM to HTML is written in Perl and named *ofm2html*. For the management of resources, a special-purpose Web browser called *MaKRV* is designed and is written C++ using Microsoft Foundation Class (CHtmlView). The idea is very simple; OFM support the LaTeX-like mathematical expression with hyper reference such as

`\href[URI]{link text}`

then MaKRV acts depening on file extension in URI and its date. MaKR-creator (*MaKRC*) is the system constructed of MaKRV and ofm2html. MaKRC manage also source codes of computer programming, that is, marking keywords and mathematical comments, the management of source codes, compile and go, etc. MaKRC is also de-

[1] *Email address: ohtsuka@hkg.ac.jp*

signed for help engine of FreeFem++[3] developed by O. Pironneau, F. Hecht and their team at the Laboratoire Jacques–Louis Lions, in Paris VI. The author is cooperating in writing the manual of FreeFem++.

For safety and copyright protection, MaKRV can access only the site (called *MaKR-site*) activated by a personal Web server working inside personal LAN, where the access from outside is denied. However we need mechanism to publish outside. We call by *pubMaK* published Website maintained by MaKRC . There are two pubMaK; One is CoMFoS, **Co**ntinuum **M**echanics **F**cusing **o**n **S**ingularities which is one of the activity groups in The Japan Society for Industrial and Applied Mathematics (`http://www.comfos.org/`). Another is the site for teaching located at `http://www.hkg.ac.jp/~ohtsuka`.

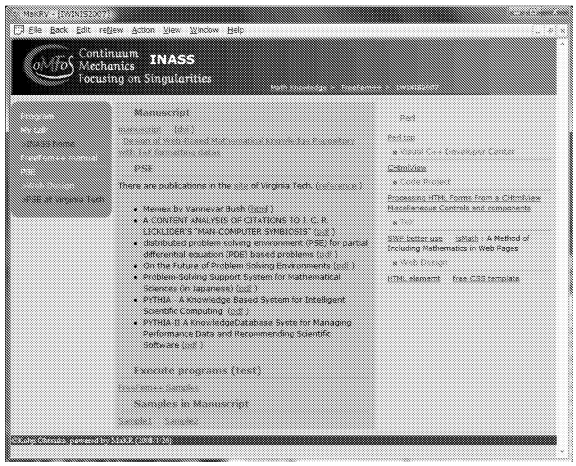

Figure 1. snapshot of MaKR-browser showing the page created from the codes in Example 1

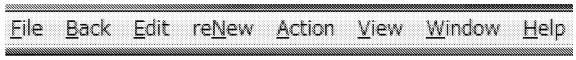

Figure 2. Menu of MaKRV

2 MaKRC

AS overview of MaKR, we start from *MaKRV* is written C++ using CHtmlView[2] which provides the functionality of the WebBrowser control within the context of MFC's document/view architecture, and use the core of the Internet Explorer application.

2.1 MaKRV

When you choose one of the menus at the top of MaKRV (see Fig. 1), the menu produces the following motion or displays a pull-down list of available commands (see Fig. 2).

File: New, open, close, print, and leave MaKRV. By [new], a child window is created and opened.

Back: Back to previous page.

Edit: Execute the favorite editor and the editor open the OFM file corresponding the page opened just now.

reNew: Create the web page if the original-file is changed, or refresh the web page if the original page is older.

Action menu: Delete, run, to TeX and rebuild. By [delete], we can delete the HTML-file and the original-file. [run] compile the original-file and go, and make new HTML-file containing the result. [to TeX] translates the original-file to TeX-file. [rebuilt] make HTML-file from the original without check of the date.

View menu: Home, current dir, favorite, HTML source, OFM source and scale. [home] open the top of MaKR. [current dir] opens the network (local) directory where the original and web page browsing just now. [favorite] opens favorite.html. [HTML source] open the web page by the editor. [OFM source] open the OFM-file by the editor. [scale] is the menu to zoom in/out of the page.

Window menu: Open, arrange windows and split windows.

Help: Under construction.

2.2 LaTeX-like formatting language OFM

Any word starting with a backslash '\' is given a special interpretation when *ofm2html* reads it from the input file. Such a word is called a *control sequence*. There are two types of control sequences: an *escape character* is a backslash followed by letters terminated by the space characters (space, tab, newline, etc.), and a *control-word* is a backslash followed by letters which con-

trol the *group* starting by the character '{' and terminated by the character '}'. MaKRC translate escape characters to readable HTML output by changing escape characters to greek and other special character codes for HTML, for example, the escape characters

> \alpha and \to are translated to α (α) and → ($\rightarrow$),

respectively. The example of the control-word is sectioning command

\section{Introduction}

where the control-word \section control 'introduction' inside the group which determine the family and size of letters, and MaKRC also make menu entries in Web page from sectioning commands. The control-word also have the option if necessary, for example, the hyper-reference is expressed as

```
\href[inass/index.html]{INASS}
```

and is translated to HTML-command

```
<a href="inass/index.html">INASS</a>
```

We call by *environment* the sentence opening and closing control-words \begin and \end, respectively, for example, the special form

\begin{document} . . . \end{document}

provides the body of OFM file. Groups and environments may be nested arbitrarily, which provides us with our hierarchy. However, they must never overlap. To display mathematical symbols, there is math mode enclosed by $ signs. The double dollar sign ($$) is used to indicate the beginning and end of display math mode such as

```
Consider Dirichlet integral defined on
a domain $\Omega$,
$$\int_{\Omega}|\nabla u|^2dx$$
```

which must be show as follows

Consider Dirichlet integral defined on a domain Ω,

$$\int_{\Omega} |\nabla u|^2 dx$$

Example 1 (Sample codes of OFM) *The following OFM code is the outline of the file* inass/index.ofm *that create the Web page shown in Fig. 1.*

```
\documentstyle{MaKRindex}
\title{INASS}
\begin{document}
\begin{header}
\section{\IMG[align="left"]{MaKR.gif} INASS}
\begin{navi}
\item \href[..]{Math Knowledge}&#65310;
\item \href[../index.html]{FreeFem++} &#65310;
\item INASS
\end{navi}
\end{header}
\begin{sidemenu}
\begin{itemize}
\item\see[http://www.inass.org/]{>INASS home}
\item \href[..]{FreeFem++ manual}
\item\href[..PSE/index.html]{PSE}
\item \see[..]{>Web Design}
\item \see[..]{>PSE at Virginia Tech}
\end{itemize}
\end{sidemenu}
\begin{main}
\begin{about}
\subsection{Manuscript}
\href[inass-b4.tex]{manuscript beta4}
\qquad (\href[inass-b4.dvi]{dvi}) ( omission )
\end{about}
\begin{info}
\subsection{Perl}
\href[..]{Perl top}
\subsubsection[class=title1]{ ■ Visual C++}
\see[..]{CHtmlView}
\ subsubsection[class=title1]{ \ Code Project}
( omisson )
\end{info}
\end{main}
\end{document}
```

The preamble is a collection of commands that specify the global processing parameters placed forward of the body.

2.2.1 Preamble

A document prepared in OFM source usually begins with a documentstyle command. For example,

\documentstyle{*doc-type*}

is used to begin a document prepared for HTML with CSS determined by *doc-type*. The *doc-type* specify the document's layout, the navigator and the sidemenu/menu (see Fig. 3) which is created from automatically from *section*, *subsection,* *subsubsection* for some *doc-type*. The following *doc-type* are implemented

MaKR make HTML with the style sheet MaKR.css,

MaKRindex make the main Web page of a Web-site of a group as in Fig. 1 (left-hand side in Fig. 3),

MaKRlinks make Web page with the sidemenu made automatically from *section*, *subsec tion* (left-hand side in Fig. 3),

article, jarticle emulate the corresponding feature of LaTeX with the style sheet designed referring to the section 13 in the book[4] (right-hand side in Fig. 3) and has the table of contents made from *section*, *subsection,* *subsubsection*

MaKRdiary make the skeleton of the diary with the time-stamp,

slides make a collection of slides fitting the display size 1024×768 with the buttons; next slides, previous slides and the table of contents made from *section*, *subsection*,

slides640 make slides fitting VGA(640×480) and used tor help of MaKR.

The other preambles are listed below.

\private{*SiteName*}: By *SiteName*, MaKRC set the place where CSS files, common images, Java scripts exist, for example, if SiteName is CoMFoS, then the Web page use CSS file and images in

`http://www.comfos.org/`

\original{*filename*}: When MaKRV accesses the file *something.type* (*type* is the extensions, edp, c, cpp, css, JSP, pl, so on), MaKRV make new OFM file *something.ofm* and translate it to HTML file *something.html* automatically. This preamble is used to memorize the original filename *something.type*.

\bakePlace{*html*}: This also created automatically by MaKRV to memorize the return page.

\js{*filename*}: This is used to include the external JavaScript file located in *SiteName/js/*.

similar to TeX: \title, \date, \author, \thanks.

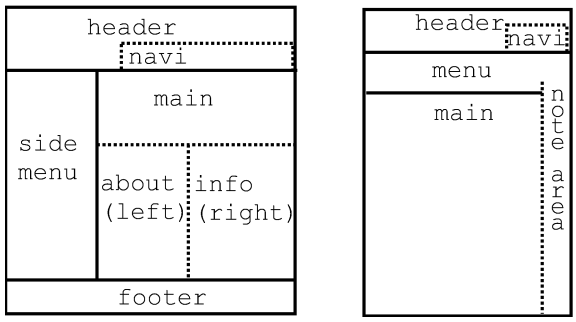

Figure 3. Standard layout created from OFM-files

2.3 Environment

MaKR add new environments listed below (see Example1 and Fig. 1).

header design the area *header* in Fig. 3.

navi make the list of URIs located in *navi* in Fig. 3.

sidemenu make the list of URIs located in *sidemenu* or *menu* in Fig. 3.

main give the environment inside the area named by *main* in Fig. 3.

footer give the environment inside the area named by *footer* in Fig. 3.

about is prepared to write the informations of a Web-site of a group.

info is prepared to write the related topics.

These environments are corresponding to the CSS ID selector. If the *doc-type* is one of article, jartilce, slides and slides640, then MaKRC make header, navi and menu and enclose the body by the environment *main* automatically.

By setting the *type* of programming language C, Cpp, ffempp, Java, JSP, HTML, Perl and XSL, we can mark the region by the form

`\begin{`*type*`}` . . . `\end{`*type*`}`

which is divided to areas of source code and comments. The supported programming languages are C, C++, FreeFem++, Java, JSP, HTML, Perl and Extensible Stylesheet Language(XSL). In the area of source code, the keywords are highlighted and we can write the TeX-like mathematical expression inside of comment area.

Example 2 *The environment for FreeFem++.*

```
\begin{ffempp}
/* This solve the problem numerically
$$
-\Delta u = f\qquad
\textrm{in}\quad \Omega; \qquad
 \partial u/\partial n=0\qquad
\textrm{on}\quad \partial\Omega
$$
where $\Omega$ is the square domain
$]0,1[\times ]0,1[$. */
 mesh Th = square(20,20);  // make
      //triangulation $\mathcal{T}_h$
fespace Vh(Th,P1);// the finite element
      // space defined over $\mathcal{T}_h$
     // is named here Vh
\end{ffempp}
```

Here's what the output of Example looks like:

/* This solve the following problem numerically

$-\Delta u = f \quad \text{in} \quad \Omega; \qquad \partial u/\partial n = 0 \quad \text{on} \quad \partial\Omega$

where Ω is the square domain $]0,1[\times]0,1[$. */
`mesh Th = square(20,20); // make`
`//` triangulation $\mathcal{T}_h$
`fespace Vh(Th,P1); // the finite element`
`//` space defined over $\mathcal{T}_h$
`//` is called here Vh

2.4 Hyper references

A Web reference accessing URI as in HTML is

\href[*URI*]{link text}

The MaKRV ignores URIs outside MaKR-site for safety, because MaKRV takes action according to the extension. For accessing the external URI, there is the command

\see[*URI*]{link text}

which open new window by the usual browser and move to URI. Perhaps, the usual browser will be safety. If *link text* inside \href is clicked, then MaKRV acts in the following manner:

html: In the case that the extension of filename is *html*;

1. If target HTML-fille isn't, MaKRV open the selection page with the following list:
   - MaKR template
   - index page (MaKRindex)
   - collection of links (MaKRlinks)
   - Japanese articles and short papers (jarticle)
   - English articles and short papers (article)
   - diary (MaKRdiary). MaKRC create the skeleton page with date-stamp.
   - presentation (slides)
   - small slides and help (slides640)

   After selection, MaKRV make OFM file and convert it to HTML file. By pushing [Edit] in the menu, we can edit the skeleton by the favorite editor. After editing, push [reNew], then new HTML-file will be created.
2. If HTML-file exists without corresponding OFM-file, then MaKRV open it without making OFM.
3. In case the corresponding OFM-file exists without HTML, OFM-file will be translated to HTML and will be opened.
4. If HTML-file and corresponding OFM-file exist. If OFM is newer, then OFM-file will be translated to HTML-file and is opened. If OFM is older, then MaKRV only open HTML-file.

edp, c, cpp, java, css, pl, jsp, xsl: If there is the source code file *somthing.type* without corresponding OFM-file, then MaKRV make a temporary file to make OFM formatting file with the structure

```
\documentstyle{source}
\title{somthing.xxx}
\original{somthing.xxx}
\backPlace{return page}
\begin{document}
\begin{type}
... ...
\end{type}
\end{document}
```

and translate it to HTML. By pushing [run] in the menu [Action], we can compile the source code and go.

doc, xls, pdf, tex: execute the associate application.

bat: A batch file *filename.bat* run by
cmd.exe /C*filename.bat* > temporary file
and MaKRV open *bat_tmp.html* created from the temporary file.

3 Low level OFM format

We use notations; $\backslash\alpha$ the control-word, β the option and A the word. The the sequence A_1A_2, consisting of two words A_1, A_2, stands for the sentence, that is A_i in A_1A_2 $(i = 1, 2)$ are delimited by the space characters (space, tab, newline, etc.), comma and period, for example A_1 ="my", A_2 ="home" make the sentence A_1A_2 ="my home". By O we denote a sequence of escape characters, control-word with group, and the sentence. We can write the fundamental structure in OFM format by

escape character: $\backslash\alpha$

control-word:

$$\alpha|\beta\{O_1O_2\cdots O_J\},$$
$$\backslash\alpha\{O_1O_2\cdots O_J\} =\backslash\alpha|\varepsilon\{O_1O_2\cdots O_J\}$$

where ε stands for empty.

We also write an **environment** as

$$\backslash\mathbf{begin}\{\alpha\}[\beta]O_1O_2\cdots O_J\backslash\mathbf{end}\{\alpha\},$$

which equivalent to

$$\backslash\alpha|\beta\{O_1O_2\cdots O_J\}$$

moreover there are exceptionable structures in LaTeX, for example \frac{1}{2}=$\frac{1}{2}$ and the table

This	is	a sample
of	the	table,

whose TeX format is

```
\begin{tabular}{|c|c|c|}
\hline
This & is & a sample \\
\heline
of & the & table,
\hline
\end{tabular}
```

Another exception is '$' mark, that is, $\$O_1O_2\cdots O_J\$$ means $O_1O_2\cdots O_J$ is in math mode and $\$\$O_1O_2\cdots O_J\$\$$ means $O_1O_2\cdots O_J$ is in display math mode.

3.1 Translation from OFM to low-OFM

3.1.1 Preamble

The preamble is a collection of commands that specify the global processing parameters, such as paper layout, title, address, the global resource, so on. ofm2html cut out a parameter from

```
\control-word{Parameter}
```

and store *Parameter* into the hash `%info`

```
%info = (
  '\documentstyle', "jarticle",
  '\title', "Untitled",
  '\css', "non", '\ajax',"non",
  ...  ...  ...
  '\mathml', "off", '\date', "a",
  '\thanks', "a", '\original', "a",
  '\res', "a", '\base', "a",
  '\backPlace', "a", '\docRoot', "a"
);
```

3.1.2 Body

Low level OFM format (abbr. low-OFM) has only two types, that is, escape character and control-word

$$\backslash\alpha|\beta\{O_1O_2\cdots O_J\backslash\}$$

The first pass in the translation from OFM to HTML is to create the file *tmp.$$$* written in *low-OFM*. The formatting language low-OFM has been designed three years before, and OFM is designed after low-OFM.

Now we explain the processes in first pass:

Elimination of newline: Eliminate newline characters inside $O_1O_2\cdots O_J$.

Metacharacter: The metacharacters $, ^, _ have the special meaning such as x^2 is x^2 and a_{ij} a_{ij}. They are translated as follows:

$$\begin{aligned} \$O_1O_2\cdots O_J\$ &\rightarrow \texttt{\textbackslash math}\{O_1O_2\cdots O_J\backslash\}, \\ \$\$O_1O_2\cdots O_J\$\$ &\rightarrow \\ & \quad \texttt{\textbackslash displaymath}\{O_1O_2\cdots O_J\backslash\} \\ O_1\hat{}O_2 &\rightarrow O_1\backslash\mathrm{sup}\{O_2\}, \\ O_1\_O_2 &\rightarrow O_1\backslash\mathrm{sub}\{O_2\} \end{aligned}$$

In TeX, the standard math text is shown in italic, numbers are roman and operators has small space +,-,×,= in the front/rear of operators. Then the pattern A =[0-9]+ in math mode is grouped as \mn{A\} and the pattern A =[\+\=\-\/] as \mo{A\}.

Order of grouping: The translator processes from a wide part to a narrow part, that is

$$\begin{aligned} & \backslash\alpha_1|\beta_1\{O_1\backslash\alpha_2|\beta_2\{P_1\cdots P_k\}O_2\cdots O_J\} \\ \rightarrow\ & \backslash\alpha_1|\beta_1\{O_1'\backslash\alpha_2|\beta_2\{P_1\cdots P_k\}O_2'\cdots O_J'\backslash\} \\ \rightarrow\ & \backslash\alpha_1|\beta_1\{O_1'\backslash\alpha_2|\beta_2\{P_1'\cdots P_k'\backslash\}O_2'\cdots O_J'\backslash\} \end{aligned}$$

Before the process, the characters '{', '}' are changed to \lbrace and \rbrace, respectively.

Fraction: \frac{O}{P} →\frac{{O}{P}\} →\frac{\mune{O'/}\deno{P'/}\} and is translated in the second pass to
<table><tr>
<td class="mune">O^H</td>
</tr><tr>
<td class="deno">P^H</td>
</tr></table>
where O^H and P^H are HTML-codes converted from O and P.

Table: The most difficult translation is the table. Moreover, we permit the mathematical expression inside a table. In the table, the row is delimited by '\\', column is split by '&', that is,
\begin{tabular}{attributes}
O_{11}&$\cdots$&O_{1n}\\$\ddots$\\$O_{m1}$&$\cdots$&$O_{mn}$
\end{tabular}
The translator first memorize the attributes γ_i ($i = 1, \cdots, m$), read elements until \end{tabular} and split the elements O_{ij}. Each row is changed to \row{O_{i1}&$\cdots$&O_{im}\} and to \row{\col|γ_1{O_{i1}\}$\cdots$\col|γ_m{O_{im}\}\}

Storage: If \α{O} (α is section, subsection, subsubsection), then O is stored to the hash %secList as
$secList{$secNum} = O
where $secNum is the numbering $m-n-l$ with the order m of section, n subsection and l subsection. That are used in making the submenu/menu.

We now explain how to translate the following;

Solve the equation $x^2+x+1=0$.

First, '$' is changed to '\math{' and search the next '$' and store the sentence x^2+x+1=0 to the buffer *inside*. Putting
%supSub = ('^','\sup{','_','\sub{'),
we use the substitute commands

```
$inside =~
    s/([\_\^])\{/$supSub{$1}/gc;
$inside =~
    s/([\+\=\-\/])+/\\mo\{$1\}/g;
$inside =~
    s/([0-9\[\]\(\)]+)/\\mn\{$1\}/g;
```

After the substitution, we get '\math{'+$inside+'\}' and write it to *tmp.$$$*.

3.1.3 Environment

We can write an environment as

$$\backslash begin\{\alpha\}[\beta]O_1O_2\cdots O_J\backslash end\{\alpha\}$$

, which is translated to $\backslash\alpha|\beta\{O_1O_2\cdots O_J\backslash\}$ in the first pass. When the analyzer meet the sentence $\backslash begin\{\alpha\}[\beta]A_1$, $\backslash a|\beta\{A_1$ is written in buffer. During searching $\backslash end\{\alpha\}$ and putting $O_1, \cdots, O_J$ into the buffer, the newline characters are eliminated except when α is the type of programming language (ffempp, C++, C, etc.). When α is the type of programming language, the analyzer divide $O_1O_2\cdots O_J$ to the front S_1 of a comment, the comment C_1 and the rear B_1 of S_1C_1, that is,

$$O_1O_2\cdots O_J \rightarrow S_1C_1B_1$$

and mark the keywords inside S_1. For C_1, we translate it as LaTeX-like. Next,

$$B_1 \rightarrow S_2C_2B_2$$

3.2 Translation from low-OFM to HTML

Here after, we denote by O^H the corresponding HTML-code of O. The analyzer of low-OFM watches the patterns

$$F^H\backslash\alpha|\beta\{O_1O_2,\cdots O_J\backslash\}N$$

where F^H is HTML-code in front of $\backslash\alpha|\beta\{O_1O_2,\cdots O_J\backslash\}$ and N is low-OFM at the rear. When the analyzer find $\backslash\alpha|\beta\{O_1$, the analyzer push α and β to the keyword stack Γ_k and the option stack Γ_o, respectively and change the sentence O_1 to O_1^H determined by (α,β). Next, check the start position p of O_1^H and push p to the position stack Γ_p. Now we put $\Gamma=(\Gamma_k,\Gamma_o,\Gamma_p)$. When $\backslash\}$ is found, the analyzer pop $\Gamma\rightarrow(\alpha,\beta,p)$ and by (α,β,p), $O_1^HO_2\cdots O_J\rightarrow O_1^HO_2^H\cdots O_J^H$. If $\Gamma=\varepsilon$, then $F^HO_1^HO_2^H\cdots O_J^H$ is written to HTML-file. Otherwise, the new sentence $F^HO_1^HO_2^H\cdots O_J^HN$ remain. We illustrate it in the case $F^H,N=\varepsilon$

$$\backslash\texttt{href}|\texttt{index.html}\{\texttt{top of my home}\backslash\}$$

In this case, $(\alpha,\beta,p)=$ (`href`, `index.html`,0) and $O_1=$‘`top’`, $O_2=A_1A_2A_3$, $A_1=$‘`of’`, $A_2=$‘`my’`, $A_3=$‘`home’`, `and` (α,β) add ‘<a href="index.html">’ to A_1, that is $O_1^H=$‘`top`’. When the analyzer find $\backslash\}$, the analyzer pop $\Gamma\rightarrow(\alpha,\beta,0)$, and add ‘’ to the tail of $O_1^HA_1A_2A_3$, that is, $O_1^HA_1A_2A_3$ become to $O_1^HA_1A_2O_2^H=$‘top of my inass’. Since the stack Γ is empty, $O_1^HA_1A_2O_2^H$ is written to the tail of HTML-file.

Next we consider the pattern

$$\backslash\alpha_1|\beta_1\{O_1O_2\backslash\alpha_2|\beta_2\{P_1\backslash\alpha_3|\beta_3\{Q_1Q_2\backslash\}P_2\backslash\}O_3\backslash\}$$

1. When $\backslash\alpha_1|\beta_1\{O_1$ is found, $\backslash\alpha_1|\beta_1\{O_1\rightarrow O_1^H$, check the start position p_1 of O_1^H and push
   $(\alpha_1,\beta_1,p_1)\rightarrow\Gamma$.
2. When $\backslash\alpha_2|\beta_2\{P_1$ is found, $\backslash\alpha_2|\beta_2\{P_1\rightarrow P_1^H$, check the start position p_2 of P_1^H and push
   $(\alpha_2,\beta_2,p_2)\rightarrow\Gamma$.
3. When $\backslash\alpha_3|\beta_3\{Q_1$ is found, $\backslash\alpha_3|\beta_3\{Q_1\rightarrow Q_1^H$, check the start position p_3 of Q_1^H and push
   s $(\alpha_3,\beta_3,p_3)\rightarrow\Gamma$.
4. When $\backslash\}$ is found, $\Gamma\rightarrow(\alpha_3,\beta_3,p_3)$, get the sequence $Q_1^HQ_2$ by p_3 and $Q_1^HQ_2\rightarrow Q_1^HQ_2^H$ by (α_3,β_3).
5. When $\backslash\}$ is found, $\Gamma\rightarrow(\alpha_2,\beta_2,p_2)$, get the sequence $P_1^HQ_1^HQ_2^HP_2$ by p_2 and $P_1^HQ_1^HQ_2^HP_2\rightarrow P_1^HQ_1^HQ_2^HP_2^H$ by (α_2,β_2).
6. When $\backslash\}$ is found, $\Gamma\rightarrow(\alpha_1,\beta_1,p_1)$, get the sequence $O_1^HO_2P_1^HQ_1^HQ_2^HP_2^HO_3$ by p_1 and $O_1^HO_2P_1^HQ_1^HQ_2^HP_2^HO_3\rightarrow O_1^HO_2^HP_1^HQ_1^HQ_2^HP_2^HO_3^H$ by (α_1,β_1).

In the second path in *ofm2html*, *ofm2html* reads a line of strings from *tmp.$$$* and search the patterns $\backslash\alpha\{A$, $\backslash\alpha|\beta\{O$ or $\backslash\}$. If the pattern $\backslash\alpha|\beta\{O$ is found, then *ofm2html* call the subroutine *startOp*(α,β) and check p in *startOp*(α,β) and push (α,β,p) to the arrays @keyList, @optList and @gsPosList, respectively. If $\backslash\}$ is found, *ofm2html* pop (α',β',p') from

(@keyList,@optList,@gsPosList)

and call the subroutine *endOp*(α',β',p'). The skeleton of the procedure above is the following:

```
while (m/(\\([a-zA-Z]+)\{
    |\\([a-zA-Z]+)\|([^\{]+)\{
              |(\\[\}]))/cg) {
  if ($2 eq undef) {
    $getkey = $3; $getopt = $4;
  }
  else { $getkey = $2; $getopt = $3; }
  $gs_pos = pos($_);
  unless ($getkey eq undef) {
    push(@keylist,$getkey);
    push(@optlist,$getopt);
    &startOp($getkey);
    push(@gsPosList,$gs_pos);
  }
  else {
    $popkey = pop(@keylist);
    $popopt = pop(@optlist);
    $old_gs_pos = pop(@gsPosList);
    &endOp($popkey,$popopt);
 }
}
```

4 Conclusion

MathML[9] step into the limelight in Web world for describing mathematical notation. But MathML

is a markup language for the browser, not for human. LaTeX keep the structure of mathematical notation comparing with MathML. OFM does not support LaTeX-feature fully just now, because it is very difficult to display mathematical expression only by HTML codes, so it will be important to make translator from OFM to MathML.

OFM formatting documents are difficult to edit as well as LaTeX, which will help WYSIWYG LaTeX editors[6, 7]. But we must write an interface to OFM (low-OFM). The prototype is built by myself and is designed from the need for the tool covering a wide field (official work, mathematics, programming, solid mechanics, finite element method, so on). At the moment, MaKRC cover the basic need and must be simple for easy modification and extension.

MaKR and CoMFoS are partially supported by the Ministry of Education, Grant-in-Aid for Scientific Research(B), 1934007, 2007 and by Hiroshima Kokusai Gakuin University from 2005 to 2006.

References

[1] American Mathematical Society,“ Math on the Web”, *http://www.ams.org/mathweb/*

[2] Microsoft, MSDN library, *http://msdn2.microsoft.com/en-us/library/*

[3] F. Hecht, O. Pironneau, A. Hyaric, K. Ohtsuka, “FreeFem++Manual”, *http://www.freefem.org/ff++*, 2007.

[4] E. Meyer, “Eric Meyer on CSS. Mastering the language of Web Design”, New Riders Pub, 2002.

[5] J. Licklider, “IRE Transactions on Human Factors in Electronics”, HFE-1, pp.4–11, 1960.

[6] MacKichan Software, Inc., Scientific Word and Scientific WorkPlace, *http://www.mackichan.com/*

[7] Design Science, Inc., *http://www.dessci.com/en/products/mathtype/*

[8] R. Mayans, “The Future of Mathematical Text: A Proposal for a New Internet Hypertext for Mathematics”, *Jour. Digital Information*, Vol.5 , Article No. 234, 2004.

[9] W3C, “Mathematical Markup Language (MathML) Version 2.0”, *http://www.w3.org/TR/MathML2/*, 2003.