Improving Accessibility to Mathematical Formulas: The Wikipedia Math Accessor*

Leo Ferres
Department of Computer Science
University of Concepción
Concepción, Chile
leo@inf.udec.cl

Jose Fuentes Sepúlveda
Department of Computer Science
University of Concepción
Concepción, Chile
jfuentess@udec.cl

ABSTRACT

Mathematics accessibility is an important topic for inclusive education. We tackle the problem of accessing a large repository of mathematical formulas, by providing a natural language description of the more than 350,000 Wikipedia formulas using a well-researched sub-language targetting Spanish speakers, for whom assistive technologies, particularly domain-specific technologies like the one described here, are scarce.

Categories and Subject Descriptors

H.5.2 [**INFORMATION INTERFACES AND PRESENTATION**]: Natural Language; I.2.1 [**ARTIFICIAL INTELLIGENCE**]: Applications and Expert Systems—*Natural language interfaces*

General Terms

Human Factors

Keywords

Accessibility (Blind and Visually Impaired), Natural language Interaction, Mathematical formulas

1. INTRODUCTION

Criticism of biases and inconsistencies aside, Wikipedia is the largest free encyclopedia, featuring over 17 million articles in total, 3.5 million in English and over 650,000 articles in Spanish[1]. Wikipedia is mostly text-based, and accessing its *textual* information through screen-readers is, relatively speaking (relative to PDFs, for instance), quite straightforward. However, Wikipedia also features images (static and animated `gif` files), sound and other multimedia content, which is, perhaps not surprisingly, much less accessible to less-abled communities such as blind and visually-impaired, or deaf people.

One such problematic content type, and specifically for the blind community, is mathematical formulas. In Wikipedia, there are (as of the date of this writing) a total of 355,684 mathematical formulas spread over 26,174 Wikipedia articles. The mathematical formulas in Wikipedia are displayed using rasterized images (`png` files) of LaTeX expressions embedded in the `<alt>` attribute of an `<img>` HTML tag, where for, for example[2],

$$\gamma = \frac{1}{\sqrt{1 - \frac{v^2}{c^2}}} \tag{1}$$

will be encoded along the lines of

```
<img
  class="tex"
  alt="\gamma =\frac{1}{\sqrt{1-\frac{v^2}{c^2}}}"
  src="name.png"
>
```

Images are then hardly accessibly (the `<alt>` tag is used for writing the LaTeX source), and LaTeX itself is both hard to understand on its own (hard to follow when read out loud), and not necessarily known by the content seeker. Thus, the problem reduces to how we can efficiently help the visually-impaired community access mathematical formulas from Wikipedia articles using their own screen-reader technology.

In this work, one more issue proved relevant: most of the natural language-based accessibility being done in the world (perhaps for funding reasons) is in English. We mean to be inclusive of other languages in this paper, and in particular, since this work originates in Chile, of the Spanish language, according to the Ethnologue[3] the second most widely-spoken language in the world, with 329 million native speakers.

This paper introduces the creatively named MathAcc, an Assistive Technology (AT) designed to help visually impaired people gain access to graphical representation of mathematical formulas published in Wikipedia, using their own screen-readers. We provide an improved language description (in Spanish) of the otherwise purely graphical information contained in formulas. The result is an application which is a significant improvement over the current ATs,

*A full version of this paper is available as *Improving Accessibility to Mathematical Formulas* at `http://www.inf.udec.cl/~leo/w4a2011.pdf`

[1] `http://stats.wikimedia.org/EN/Sitemap.htm`

[2] `http://en.wikipedia.org/wiki/Special_relativity`

[3] `http://www.ethnologue.com/`

W4A '11, Mar 28-29 2011, Hyderabad, Andhra Pradesh, India.

as it allows users to explore and interact with Wikipedia formulas.

2. BACKGROUND AND RELATED WORK

2.1 A Bird's Eye View of Accessibility, Blindness and Education in Chile

The CASEN (Caracterización Económica Nacional) survey, applied by the Ministry of Planning is a comprehensive national survey of the socio-economic context of the country [7, 8]. The first CASEN was carried out in 1985, while the latest one dealing with disabilities was in 2006. In this 2006 incarnation of the survey, 73,720 homes were surveyed, and a total of 288,873 people were represented in the data. In all, 6.9% of this sample had a "long-lasting or severe disability", of which 45.6% were due to blindness. An important point is that disabilities in Chile impact very negatively in the educational level achieved by disabled communities: only 43% of people in these communities finish "Educación Básica" (tr. Basic Education, or education people receive until they are 12 years old.)

Similarly, Chile administers the PSU (Prueba de Selección Universitaria, University Selection Test), very similar to the United States SAT Reasoning Test, and this constitutes the admission system to all Chilean universities. The test consists of two mandatory exams, mathematics and language (and other specific tests that certain departments may require for admission).

In policy making, Chile is one of the relatively few countries with a full-fledged, up-to-date law (Ley 20422[4]) dealing with accessibility, social inclusion and equal opportunities for people with disabilies. This law was sanctioned on February 10, 2010, and highlights the need for accessibility to processes, tools and services (Article 3). In the framework of this law, the PSU was administered in Braille for the first time in December 2010.

2.2 Available technologies for math accessibility

Recently, researchers have become very interested in the problem of accessibility to mathematical formulas by the blind community, and excellent quality research has been produced[5]. Among them there is the LAMBDA project [2], funded by the European Union in the Information Society Technologies[6]. The system consists of a markup language to represent mathematical expressions, not unlike, for instance, *MathML*, that can be directly translated to the 8-point Braille system[7], an extension of the 6-point system. This is an excellent choice for certain contexts, but today's availability of digital information makes the use of Braille an expensive and/or slow technology (if the formulas have to be printed/embossed paper, the cost of the printers, and the paper is quite substantial). Also expensive are the different electronic Braille technologies like keyboards or Braille displays.

AsTeR, in turn, is a system that helps produce renderized audio of electronic documents [10]. This system takes a LaTeX document and parses it, generating a tree structure representation. This parsed document is then input to a system that generates language based on rules, called *Audio Formatting Language*, or AFL. The AFL language maps the tree's nodes onto the audible portion of the node contents that allow the sound presentation of mathematical formulas. It is interesting to notice that to highlight certain nodes, modifications are done to the intonation or voice inflections and pauses between particular terms in a formula. For example, to indicate a superindex, the pitch will be higher, while for a subindex, the pitch will be lower. *AsTeR* also allows navigation of the formulas, exploring the expression as a tree-based structure (ancestors, siblings) and tagging certain nodes to easily come back to them later, etc.

Stanley and Karshmer introduce *MathGenie* [15], a system that reads out mathematical formulas, together with their Nemeth code [9]. *MathGenie* was specially designed for students in the sciences who had some visual impairment. It allows the user to navigate formulas through a simple key combination, and output information by the text-to-speech engine and Braille displays. It receives presentation *MathML* as input, focusing mostly on how formulas are presented, rather than what their semantics are.

MathPlayer is a plug-in for Microsoft Internet Explorer [13] that was designed, primarliy, for rendering a visualization of *MathML*. It now allows users to interact with mathematical expressions through two navigational modes: one based on text that reads expressions from left to right, and another based on a tree representation of the formula that allows to navigate the formula in any direction. *MathPlayer* was designed to be easily integrated with screen-readers.

In [11], Reddy and Gupta proposed a different approach: they present a method for translating *MathML* into *voiceXML*, the W3C's standard XML format for specifying interactive voice dialogues between a human and a computer[8]. In this way, a *voiceXML* interpreter will render the documents generating audio and handling voice interaction through managed dialogues with the users.

Finally, Pontelli and Abu Doush [1] propose a framework that allows to navigate mathematical expressions using the same two modalities in [13], one based on linear access, while the second based on hierarchical accessed with a tree as the underlying data structure. This time, however, the implementation was made as a plug-in for the FireFox web browser, on top of *FireVox*[9], a FireFox screen-reader system. This system also takes presentation *MathML* as input, and thus the semantics of the mathematical expressions are under-determined.

None of these papers, however, have attempted to, given some semantics (as encoded in content *MathML*), provide linguistic descriptions to a massive repository of formulas such as the ones in Wikipedia; nor have they done a thorough investigation on the *language* used to talk about formulas, and to-date, there is no system that produces Spanish descriptions of formulas. These are our contributions.

[4]For the complete text (in Spanish) of the law, see `http://www.munitel.cl/Actualidad_Legislativa/Ley_20.422.pdf`

[5]For surveys, see `http://www.cs.washington.edu/homes/cjayant/papers/MathAccessFinal.pdf`, and `http://www.utdallas.edu/~gupta/mathaccsurvey.pdf`. Unfortunately, these papers have not been formally published yet.

[6]`http://ec.europa.eu/research/fp6/index_en.cfm?p=2`

[7]Unified 8 dot Braille Code, see `http://www.8dotbraille.com/`

[8]`http://www.voicexml.org/`

[9]`http://www.firevox.clcworld.net/`

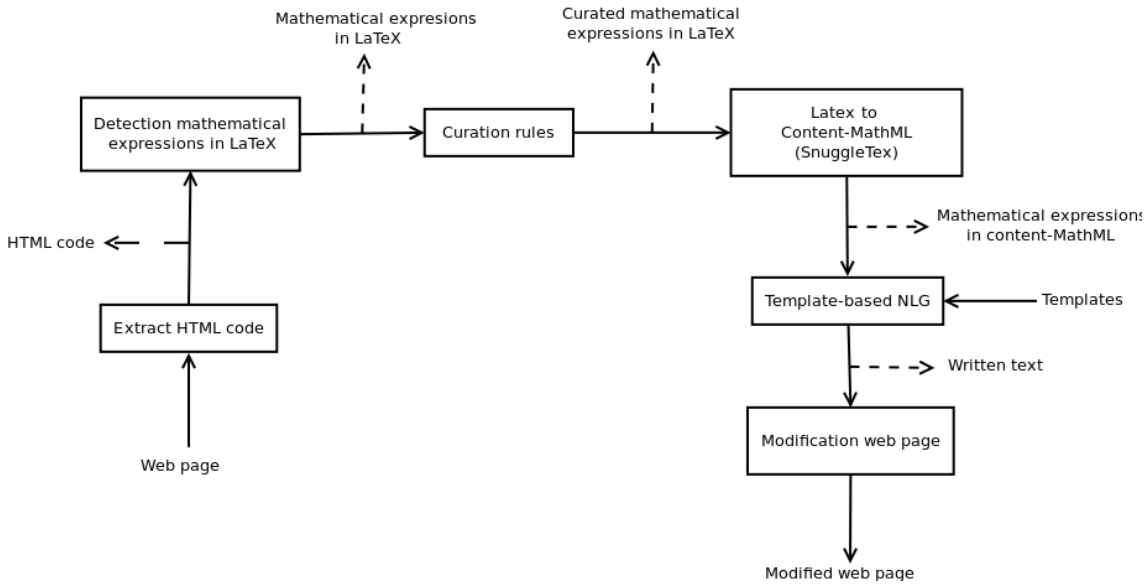

Figure 1: General architecture of MathAcc

3. SYSTEM DESCRIPTION

3.1 MathAcc architecture

The general architecture of the system is shown in Figure 1. The system consists of three sub-systems: one devoted to the detection of mathematical formulas in a Wikipedia page, together with a process that translates from LaTeX to a more structured, less ambiguous representation written in MathML (specifically *content* MathML), a template-based natural language generation system like the one in [6], and finally a module that restructures the Wikipedia page to incorporate the linguistic description of the formula. In what follows, we detail the workings of these different modules.

3.1.1 Detecting and cleaning mathematical expressions

This first module of the system is in charge of capturing the HTML code and parse it to retrieve all mathematical expressions in it. In Wikipedia articles, mathematical expressions are shown as raster images, using the `<img>` tag, with the attribute `<class="tex">`, while the attribute `<alt>` contains the LaTeX definition, as shown in the introduction (see Equation 1). Note that MathAcc does not do image processing itself, but rather works with the semantically underdetermined LaTeX string that Wikipedia exposes in the source HTML.

Wikipedia does not conform to LaTeX in a strict sense, mainly because there are technical problems in the interaction between LaTeXand HTML. This module is therefore also responsible for curating the data. The curation involved *deletion* and *substitution*, the complete list is in Table 1.

Once the LaTeX formulas were curated, the next step was to generate a more structured, less ambiguous representation. LaTeX is semantically ambiguous, in the sense that not all information about a formula is explicit in its representation. For instance, in the expression "`x^2 + px = q`", it is unclear whether p and x are two distinct variables, or a single variable called px. Likewise, there were cases in Wikipedia were the LaTeX expression "`f^{-1}`" meant either the inverse function or a variable f to the power of -1. To clean the *semantics* of the formulas, we chose to translate them into the content markup of *MathML*, see Figure 2 for an example. Mathematical Markup Language (*MathML*) is an application of XML for describing mathematical notations and capturing both its structure and content. Given the sheer volume of mathematical formulas in Wikipedia, the translation from LaTeX to *MathML* had to be done automaticall.

To do this, we used *SnuggleTex*, from the University of Edinburgh[10], a Java library that does part of the translation automatically, even though we had to curate and modify part of the LaTeX data coming from Wikipedia, among others the biconditional `iff`, which was not understood by

Deletions	`.\` (dot backslash)
	`. \` (space dot backslash)
	`\,` (backslash comma)
	`\!` (backslash exclamation mark)
	`\ ` (backslash space)
	`~` (tilde)
	`\scriptstyle`
	`\textrm`
	`\mathbf`, `\mathbb`, `\mathcal`

Substitutions	`\iff` → `\leftrightarrow`*
	`<` → `<`
	`>` → `>`
	`\ne` → `\neq`
	`\empty` → `\emptyset`
	`\tfrac` → `\frac`
	`,` → `.` (the thousand separator in Spanish)

Table 1: Curation rules. (*)The `iff` was read as a multiplication by th translator from LaTeX to *MathML*, so it had to be changed.

[10] `http://www2.ph.ed.ac.uk/snuggletex`

LaTeX	Appears	LaTeX	Appears
_	287,168	ldots	4,463
^	167,479	cdots	4,211
frac + over	77,425	log	4,187
sqrt	14,647	dots	3,629
partial	14,576	ln	3,271
rightarrow + to	13,316	otimes	2,552
sum	13,225	lim	2,464
in	12,316	choose	2,361
int	9,961	approx	2,326
times	8,428	dot	2,308
leq + le	7,327	forall	2,271
bar+overline	6,555	circ	2,153
cos	6,091	pm	2,026
sin	6,083	equiv	2,001
hat	5,866	geq	1,874
vec	4,600		

Table 2: Most frequent LaTeX keywords used in Wikipedia's formulas

SnuggleTex (see the note to `\iff` in Table 1). If we take the ambiguous example of the inverse function versus variable to the -1 power, we would get, respectively:

```
<math>                <math>
  <apply>               <apply>
    <inverse/>            <power/>
    <ci> f </ci>            <ci> f </ci>
  </apply>                  <cn> -1 </cn>
</math>                 </apply>
                      </math>
```

Figure 2: Example of content *MathML*. The left XML snippet corresponds to the inverse function, while the one to the right corresponds to the "variable f" to the power of -1.

where the semantics is made explicit by the XML *MathML* tags `<inverse>` and `<power>`, and where the tags `<ci>` denote variables, and `<cn>` denote numbers[11].

As mentioned, all in all there were 355,684 formulas, and a comprehensive count of the appearances of the LaTeX keywords are summarized in Table 2. Besides these, the system also considers greek letters, basic mathematical operations that do not have a LaTeX keyword (e.g. +, -), those which do (e.g., ×, ÷), and symbols such as the infinity symbol, ∞ among others. It is the operators in the table, plus the symbols, together with their relationships that we provide linguistic forms for (see Section 3.1.2), and a stack-based algorithm that will generate the descriptions in natural language.

Figure 3 plots the correlation between the total number of appearances of the LaTeX "symbols of interest" versus the total number of formulas they appear in. The graph shows a relatively pronounced power curve (not unlike the Zipfian curves of natural language), with subscripts ("`_`"), superscritpts ("`^`"), and fractions ("`\frac{}{}`") being the most common ones, and also the ones that are used (i.e. repeated several times) in most formulas (notable exceptions are the square root ("`\sqrt`"), and the partial ("`\partial`")).

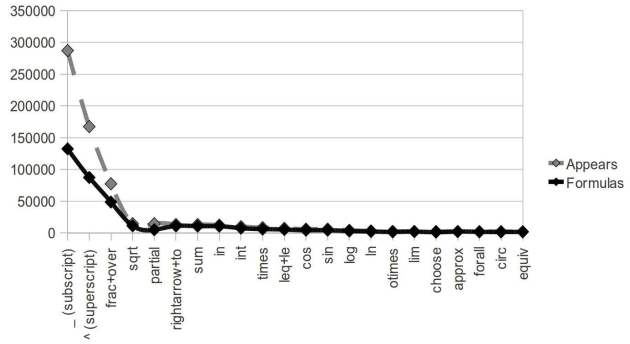

Figure 3: Correlation between the symbols and their appearance in Wikipedia formulas.

All in all, it is clear that providing a template to these concepts will in effect give an account of most of Wikipedias use of LaTeX. Obviously, this leaves aside the relations these symbols enter among each other, which, as we will see, are more difficult to formalize.

3.1.2 The Sub-Language of the Generation Module

Perhaps the largest contribution of this paper is the design of the language to be used with each *MathML* operator, which we will call the *concepts* of a formula. As explained in Section 3.1.3, ours is a template-based natural language generation (NLG) system [12, 16, 3, 14], and thus each of the concepts will be associated to a template. To find out what the best template was for each of the concepts in Table 2, we carried out simple experiment that consisted in having participants looking at a formula, and providing the best natural language description they could think of.

3.1.2.1 Methodology.

Participants. We invited sixty people through email and Facebook to collaborate in this research. Of those 60 people, we recieved a reply from 38 people. The participants came from a variety of different backgrounds, most of them from Engineering disciplines (computer engineering, industrial engineering, electronics, civil engineering), from mathematics, from education and law, and also university professors.

Stimuli. The stimuli consisted of 21 formulas coded in presentation *MathML* where the following concepts were tested: `divide` (1), `power` (1), `root` (1), `partialdiff` (1), `tendsto` (1), `sum` (2), `in` (1), `int` (2), `cartesianproduct` (1), `leq` (1), `sin` (1), `cos` (1), `log` (1), `ln` (1), `outerproduct` (1), `limit` (1), `approx` (1), `forall` (1), `compose` (1), `equivalent` (1) and `geq` (1). The reader will notice that although the overlap between the concepts in the stimuli list and those in Table 2 is high, the concepts tested are not exactly the same. Some concepts are either missing (e.g., `_` (subindex), `choose`), or they have been renamed (e.g., `root` (from the LaTeX `sqrt`), `outerproduct`, from the LaTeX `otimes`). This discrepancy was due to the use of *content MathML* as the encoding format. This choice restricted what LaTeX keywords we could use, and under what name. 68% of the most

[11] `http://www.w3.org/TR/MathML/chapter4.html#contm.cn`

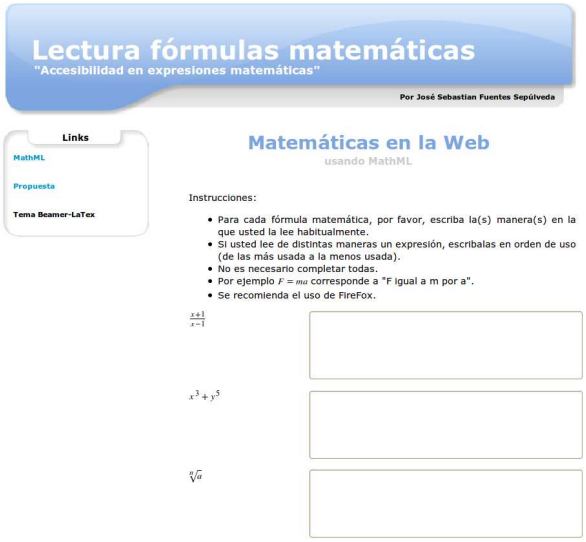

Figure 4: A portion of the website designed to present the stimuli of the study.

frequent LaTeX keywords in Table 2 is represented in the stimulus of this study.

Procedure. The formulas were presented in a webpage (see Figure 4). The answers were anonymous and were gathered in the period comprised between November 10, 2010 and December 2, 2010. The instructions, present on the test website at all times, were as follows:

> For all mathematical formulas please write the manner in which you usually read it aloud, for instance $F = ma$ could be read as "F es igual a eme por a" [tr. "Eff equals em times a"]. If you understand the formula could be read in different ways, please write them in order or preference (from the most preferred to the least). It is not mandatory to complete all of them.

3.1.2.2 Results.

We received a total of 772 answers for the 21 formulas. The answer distribution was as follows: `divide` (42), `power` (49), `root` (42), `partialdiff` (38), `tendsto` (32), `sum1` (41), `sum2` (33), `in` (43), `int1` (36), `int2` (37), `cartesianproduct` (40), `leq` (38), `sin_cos` (40), `log_ln` (37), `outerproduct` (12), `limit` (36), `approx` (36), `forall` (36), `compose` (34), `equivalent` (29), and `geq` (41).

Each of the 772 answers were grouped in terms of their semantic similarity on a first pass, and then in terms of their syntactic similarities. Thus, for example, although "x más 1, dividido en x menos 1" [tr. x plus one divided *in* [sic] x minus one] is syntactically different from "x mas 1 dividido por x menos 1" [tr. x plus one divided *by* x minus one][12], they are semantically equivalent. In turn, "x cuadrado implica a cuadrado" [tr. x square implies a square] and "x cuadrado tiende a cuadrado" [tr. x square tends to a square], mean different things.

Once the answers were grouped, we counted how many answers were in a group and how many groups there were. Except for a few special cases, the most common answer became the "official" template for that (concept) of the formula. The 21 formulas, together with the chosen template for generation are detailed in Table 3. The placeholders for the templates are as follows: OPx, where $x \in \mathbf{N}$ are *operands*, to which an operator is applied. For instance, in $x + 1$ the operands are "x" and "1", while the operator is "+". $GRADO$ [tr. degree] is the degree of a root. VAR are the independent variables of integrals, limits, etc. $LIMx$, where $x \in \mathbf{N}$, is the scope of application of a given operator, Finally, $BASE$ is the base of a logarithm.

The third column of the table gives information about what we have called the "confidence" of the chosen template. The number to the left is the percentage of answers that had the same, or a very similar syntax[13]. The number to the right is what we call the "semantic confidence" of the percentage, which is, effectively the number of semantic groups. This number gives a sense of the "dispersion" of the chosen template (the lower the number the less dispersion of its semantics, with 1 being perfect consensus).

The template of the expression x^3+y^5 does not correspond to the "majority" template. The most common template for these concepts was "OP al $GRADO$" [tr. OP to the $GRADO$ power], por ejemplo "x al cubo" [tr. "x cubed"]. However, this template was not sufficiently general, since for larger $GRADO$, the template using "al $GRADO$" would not make sense ("a la 30(?ava) potencia", [tr. to the 30th power]). Although the chosen template is more general, we have implemented both kinds, one for the common "squared" and "cubed" powers, and the less common template "elevado a" [tr. "to the power of"] for the rest.

The template of the expression $x^2 \to a^2$ does not correspond to the majority template. The most common template for this expression was "$OP1$ implica $OP2$" [tr. $OP1$ implies $OP2$]. However, it is obvious (given the form of the formula) that what we were evaluating was the "tends to" (`to` keyword) concept. We can conclude here that the keyword `to` is ambiguous when the context is not taken into account.

There was no consensus for the template of the expression $A \otimes B$. We obtained three "most common" templates, but with a very low frequency. This means that the `otimes` keyword is, contrary to intuition, highly ambiguous, and quite unknown.

Lastly, the majority template for the expression $(f \circ g)(x) = f(g(x))$ was "$OP1$ o $OP2$", however we had to discard this template due to the fact that the verbalization for the $\vee$ (logical or) operator is the same one. Thus, we adopted the second highest frequency syntactic template: "$OP1$ compuesta $OP2$" [tr. "$OP1$ composes $OP2$"].

As is usual with domain-specific sub-languages, there is marked variability in the *syntax* of the expressions, but not so in the *semantics*, except for a few outlier cases (such is the case of, for instance, $A \otimes B$). This bodes well with the methodology of "saying" mathematical information by means of screen-readers. Even if the syntax of the expressions to use in templates is a bit more diffuse, the reader should keep in mind that the differences, most of the time, had to do with changing one preposition, or one merely one other word that did not affect meaning (by definition, since

[12]The use of prepositions in Spanish is a bit looser than in English.

[13]All the data is available at `http://www.inf.udec.cl/~leo/formtemps.txt`

Formula	Template	Confidence
$\frac{x+1}{x-1}$	$OP1$ dividido por $OP2$	0.40 (1)
$x^3 + y^5$	$OP1$ elevado a $OP2$	0.33 (1)
$\sqrt[n]{a}$	raiz $GRADO$ de OP	0.55 (2)
$\frac{\partial^2 f(x,y)}{\partial x \partial y}$	[$GRADO$] derivada de OP, con respecto a VAR(, VAR)*	0.13 (6)
$x^2 \to a^2$	$OP1$ tiende a $OP2$	0.25 (4)
$\sum_{x=a}^{b} f(x)$	Sumatoria desde $LIM1$ hasta $LIM2$ de OP	0.20 (1)
$\sum_{x \in B} f(x)$	Sumatoria de OP, con DOM	0.15 (2)
$a \in A$	$OP1$ pertenece a $OP2$	0.65 (1)
$\int \sin = \cos$	Integral de OP	0.5 (1)
$\int_0^1 x^2 dx$	Integral desde $LIM1$ hasta $LIM2$ de OP respecto a VAR(, VAR)*	0.19 (1)
$A \times B$	$OP1$ cruz $OP2$	0.55 (1)
$3 \leq 3 \leq 4$	$OP1$ menor o igual que $OP2$ (menor o igual que OP)*	0.55 (1)
$\sin(\cos x + x^3)$	Seno de OP | Coseno de OP	0.60 (1)
$\log_3 x + \ln a$	Logaritmo en base $BASE$ de OP | Logaritmo natural de OP	0.65 (1)
$A \otimes B$	Producto tensorial entre $OP1$ y $OP2$ $OP1$ ex or $OP2$ $OP1$ cruz $OP2$	0.17 (8)
$\lim_{x \to 0} \sin x$	Limite de OP, cuando VAR	0.58 (1)
$\pi \approx 22/7$	$OP1$ es aproximadamente $OP2$	0.53 (2)
$\forall x : (x - x = 0)$	Para todo VAR, OP	0.53 (1)
$(f \circ g)(x) = f(g(x))$	$OP1$ compuesta $OP2$	0.24 (2)
$a \equiv \neg\neg a$	$OP1$ es equivalente a $OP2$	0.62 (3)
$4 \geq 3 \geq 3$	$OP1$ mayor o igual a $OP2$ (mayor o igual a OP)*	0.32 (2)

Table 3: Results of the semantic/syntactic grouping of the answers of the study.

it otherwise would have been grouped in another semantic cluster). Thus, choosing the "majority template" in terms of frequency is a warranted technique.

Since we now have a set of researched linguistic templates, we need to use them as input to the language generation module, which we now turn to.

3.1.3 The Language Generation Module

The backbone of the NLG system used in MathAcc has been successfully used in other applications, most notably in iGraph-Lite [5, 6, 4]. The version we are working with here, however, has a new engine that allows for stack-based generation that is vital given the nesting-based nature of mathematical expressions. In what follows, we give a detailed run of the system to explain its workings. To do this, we use, allegedly, one of the most beautiful (and universal) equations of all time: $E = mc^2$.

The whole stacktrace of the generation system is shown in Figure 5. Once the the curated *MathML* expression is input into the generation module, the XML snippet is parsed and the tree data structure we obtain is then traversed in depth-first manner, adding *MathML* elements to a stack. Every time the `apply` node is found in the tree, an `apply` node is added to the stack, signalling the start of an operator. Likewise, when a `ci` or `cn` node is found on the tree, only their children are added to the stack (see `m` on line 10 in Figure 5). In all other situations, the node is added to the stack without any modifications.

Every time the XML `apply` subtree is completely traversed (which means all the operands are now known, together with the operand itself), we start "popping" the stack until the first `apply` node to be found. The popped nodes of the stack are "pushed" into another temporary stack to be input into the subsystem that applies the templates. For instance, the first subtree of an `apply` node that has been completely traversed is `(power(ci(c), cn(2)))`, for which the stack is `['apply', 'eq', 'E', 'apply', 'times', 'm', 'apply']`, and the temporary stack contains `['2', 'c', 'power']`. Once the temporary stack has been built, the templates are applied by finding the top of the stack (last in was `power`), and searches the dictionary for the appropriate template to use. In this case, the dictionary search yields the binary operator template for `power`: "VAR elevado a VAR", where each of `2` and `c` are assigned, in "first out" order, to each "VAR". The system finally adds a `(pausa)` string at the extremes of the verbalization, obtaining the right part of line 19 in Fig. 5. Pauses are added as markers to separate semantically self-contained units in the verbalization to help text-to-speech grouping (see [10]). The process is repeated for each `apply` until the stack is empty. Two or more subsequent pauses are collapsed into one pause, and the final verbalization is complete:

```
(pausa) E es igual a  (pausa) m por  (pausa)
c elevado a 2 (pausa).
```

```
1.  []
2.  ['apply']
3.  ['apply', 'eq']
4.  ['apply', 'eq']
5.  ['apply', 'eq', 'E']
6.  ['apply', 'eq', 'E']
7.  ['apply', 'eq', 'E', 'apply']
8.  ['apply', 'eq', 'E', 'apply', 'times']
9.  ['apply', 'eq', 'E', 'apply', 'times']
10. ['apply', 'eq', 'E', 'apply', 'times', 'm']
11. ['apply', 'eq', 'E', 'apply', 'times', 'm']
12. ['apply', 'eq', 'E', 'apply', 'times', 'm', 'apply']
13. ['apply', 'eq', 'E', 'apply', 'times', 'm', 'apply', 'power']
14. ['apply', 'eq', 'E', 'apply', 'times', 'm', 'apply', 'power']
15. ['apply', 'eq', 'E', 'apply', 'times', 'm', 'apply', 'power', 'c']
16. ['apply', 'eq', 'E', 'apply', 'times', 'm', 'apply', 'power', 'c']
17. ['apply', 'eq', 'E', 'apply', 'times', 'm', 'apply', 'power', 'c', '2']
18. ['apply', 'eq', 'E', 'apply', 'times', 'm', 'apply', 'power', 'c', '2']
19. ['apply', 'eq', 'E', 'apply', 'times', 'm', ' (pausa) c elevado a 2 (pausa) ']
20. ['apply', 'eq', 'E', ' (pausa) m por  (pausa) c elevado a 2 (pausa) (pausa) ']
21. ['(pausa) E igual a (pausa) m por  (pausa) c elevado a 2 (pausa) (pausa) (pausa)']
```

Figure 5: Stacktrace of the NLG system for $E = mc^2$.

Although not part of the generator itself, the next step is to modify the wikipedia page adding the `verb` attribute with the verbalization within the `<img>` tag that contains the processed LaTeX snippet, resulting in the following modification of the website:

```
<img
  class="tex"
  src="image.png"
  verb=" (pausa) E es igual a (pausa) m por
  (pausa) c elevado a 2 (pausa)"
  alt="E = mc^2"
/>
```

In what follows we discuss a few formulas that demonstrate the power of the system. These formulas were chosen from random Wikipedia articles.

$$t = \frac{t_o}{\sqrt{1 - v^2/c^2}} \tag{2}$$

is verbalized as "(pausa) t es igual a (pausa) t sub o (pausa) dividido por (pausa) raz cuadrada de (pausa) 1 menos (pausa) v elevado a 2 (pausa) dividido por (pausa) c elevado a 2 (pausa)", [tr. "t equals t sub oh divided by the square root of one minus vee square divided by c square"]. This example shows several operators and several levels of nesting in the formula.

$$Z = \sqrt{\frac{\mu}{\varepsilon}} \tag{3}$$

is verbalized as "(pausa) Z es igual a (pausa) raiz cuadrada de (pausa) mu dividido por epsilon (pausa)" [tr. "Zed equals the square root of mu divided by epsilon], which nicely demonstrates the use of Greek letters. The next equation (equation 4) demonstrates a more common type, but it shows the multiplication of constants and variables:

$$\frac{(t+3)(t-3)}{(t+3)(2t+1)} \tag{4}$$

is verbalized as "(pausa) t más 3 (pausa) por (pausa) t menos 3 (pausa) dividido por (pausa) t ms 3 (pausa) por (pausa) 2 por t (pausa) ms 1 (pausa)" [tr. "t plus 3 times t minus 3 divided by t plus 3 times 2 times t plus 1"].

$$\xi \to \frac{\cos(\theta/2)\,\xi - \sin(\theta/2)}{\sin(\theta/2)\,\xi + \cos(\theta/2)} \tag{5}$$

is verbalized as "(pausa) xi tiende a (pausa) coseno de (pausa) teta dividido por 2 (pausa) por xi (pausa) menos (pausa) seno de (pausa) teta dividido por 2 (pausa) dividido por (pausa) seno de (pausa) teta dividido por 2 (pausa) por xi (pausa) ms (pausa) coseno de (pausa) teta dividido por 2 (pausa)" [tr. "xi tends to the cosine of theta divided by 2 times xi minus sine of theta divided by two divided by sine of theta divided by 2 times xi plus cosine of theta divided by 2"], which demostrates a long verbalization that would be difficult to generate by humans (and possibly hard to understand by us as well, but this is an empirical question), together with the use of trigonometric functions. Lastly, we show a formula for which the operator node of the *MathML* tree was left unchanged, and was not in the list of the most used ones:

$$x = 3 \cup x = 5 \tag{6}$$

is verbalized as "(pausa) x es igual a (pausa) 3 unión x (pausa) y (pausa) 3 unión x (pausa) es igual a 5 (pausa)" [tr. "x equals 3 union x, and 3 union x es igual a 5."].

The system, however has a few limitations. Most of them are due to the complexity of translating LaTeX to *MathML*, but there are other that are due to *MathML* itself. First, it is obvious that a comprehensive study of the patterns of errors obtaine for 34% of formulas for which no description was generated. To date, MathAcc can only verbalize the subset of Wikipedia formulas for which there is a LaTeX to *MathML* translation, as provided by *SnuggleTex*. The latter system does translate certain operators to content *MathML*, even if it can represent the whole expression in presentation

MathML. Examples of some of these LaTeX operators include `sum` ($\sum$), `int` ($\int$), `forall` ($\forall$), `partial` (∂), among others.

A certain subset of the expressions for which there is a verbalization is ill-formed. *Snuggletex* makes a bad translation to content *MathML*, and this translation is reflected in the verbalization. For instance, expressions in LaTeX such as x', are incorrectly verbalized as "x elevado a apostrofe" [tr. "x to the power of apostrophe"], instead of "x prima" [tr. "x prime"], $(x; y)$ is in turn verbalized as "x por punto y coma por y" [tr. "x times semi-colon times y"].

Currently, the system only understands single-letter variable names like x, y or z. When a multiple-letter variable is input, MathAcc will take this to mean a product. For instance, the expression $m = mass$ is verbalized as "eme es igual a eme por a por ese por a" [tr. "em is equal to em times a times ess time s"], still a problem we inherit from the LaTeXto *MathML* translation.

MathAcc only supports subindices with no more than one variable. For instance, `F_{x}` (F_x) or `X_i` (x_i). More complex times are not supported, such as `X_{y-z}` (X_{y-z}). Likewise, all the superindices of LaTeX expressions are translated by *SnuggleTex* as "powers of" in content *MathML*. Thus, expressions like `x^1` (x^1, or "x super 1") cannot be properly translated.

There is a subset of expressions for which there is no *MathML* translation, independent of whether it is presentation or content. A few examples include `choose` as in $\binom{n}{k}$ (the binomial coefficient), or the ellipsis `ldots` (...), which MathAcc cannot verbalize.

3.2 User's view

Although the MathAcc system has been designed, primarily, with the blind computer user in mind, the interaction with the system may be analyzed from two different points of view: the publisher of mathematical content, and the consumer of that content. In the case of the content producer, it is only necessary to add the LaTeX codification of a certain mathematical expression to the `alt` attribute to the `<img>` tag. As we will see, the system will be in charge of generating a natural language description of it.

In the case of the content seeker, and screen-reader user, the only requirement is to make the interface read the supplied `verb` attribute of math formulas in Wikipedia LaTeX. For the non-blind user, the system will show the automatic verbalization in red and boxed over the identified Wikipedia formula, as shown in Figure 6 on the next page. The user paradigm *is* simple for MathAcc.

4. EVALUATION

MathAcc is undergoing evaluation with the blind community in Concepción, but there is yet no "hard" (statistical) data about the usability of the system. However, before the usability studies can be carried out, there was the need to know how comprehensive the generation stage was. In other words, it was important to see whether too few formulas from Wikipedia were generated, or whether the generation was wrong. To get a feeling of just how MathAcc was performing, we tested the generation engine on a subset of 100, 500, and 1000 random formulas from Wikipedia[14].

On a first pass, we counted the number of times the generator could not assign a description to a formula. Quite surprisingly, very similar results were found. For the sets of 100, 500 and 1000 test formulas, the generator provided a linguistic description (either right or wrong, see below), for 66%, 65.5% and 66.1%, respectively. This is without any fine tuning, just taking the LaTeX formula, translating it into *MathML* and finally going through the generator. The reasons for the failure to generate were varied but they all had to do with how *SnuggleTex* would convert from LaTeX to *MathML*. We do not use the presentation portion of *MathML* but the content portion, and this is still underdetermined in either *SnuggleTex* or *MathML* itself, so errors there are to be found.

On a second pass, for the 66 formulas that were given a description in the 100 random formula file, we counted the number of times the description provided was wrong. Since the percentages were so close, the generation algorithm so deterministic, and pending a much more comprehensive evaluation of MathAcc, we only analyzed these 66 formulas. We hypothesize that the other test cases (500 and 1000 formula files) will be similar. Of the 66 formulas that were given a description, 6 (~10%) were wrong. Once again, this is not a bad result considering that there has been no optimization done to the translation between LaTeX to *MathML*.

Needless to say, even if the coverage of the formulas is acceptable (at least for a first pass), the sub-language used to provide descriptions should be investigated further, particularly with congenitally blind people. This is left now for

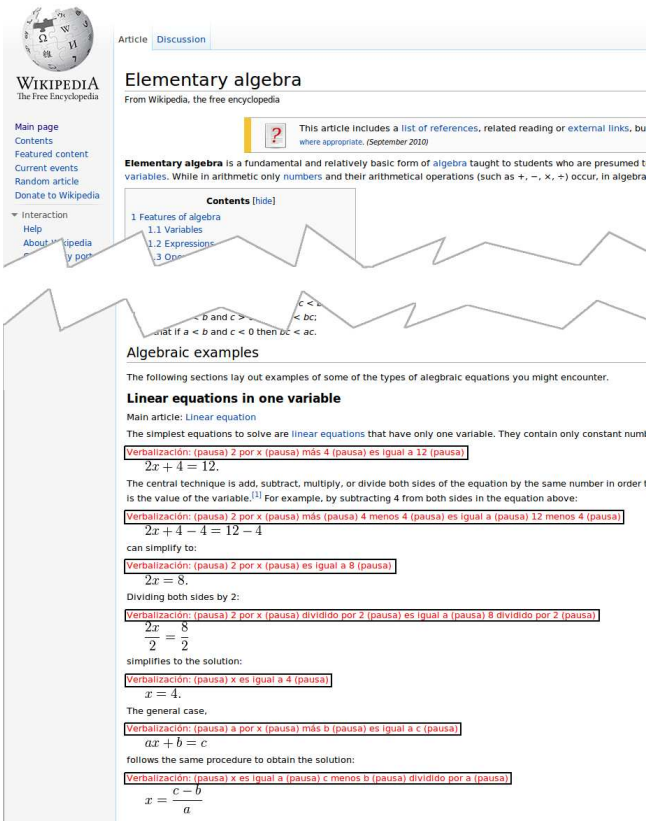

Figure 6: A screenshot of how Wikipedia looks after the MathAcc processing. The red/gray-boxed text is the content of the `verb` attribute of the `<img>` tag.

[14]The interested reader can find the results at `http://www.inf.udec.cl/~leo/formulas.html`

future work.

5. CONCLUSIONS AND FUTURE WORK

We have demonstrated that it is possible to provide rich linguistic verbalizations to *sui generis* mathematical formulas in Wikipedia, in the Spanish language, and using content *MathML* as the input to the generator (previous curation of the Wikipedia LaTeX conventions.

A few things can be improved: there is obviously the need for a better LaTeX→*MathML* translator. *SnuggleTex* is the only one currently in existence, and as we saw before, it can be made more stable and less idiosyncratic in the translation decisions. Relatedly, it would be interesting to try other formats for representing formulas like *OpenMath*[15].

It would be of interest to go deeper into the semantics of formulas, and provide a context for variables: thus, if the expression is $F = mg$, to preprocess the formula to contain the fact that m denotes mass and g denotes gravity, while F is force.

We are also looing into interfacing MathAcc with WebAnywhere[16] in the same way we did with the iGraph plugin, see for example [4]. This way, the representation of MathAcc could be accessible to blind people who are away from their computers or at a computer without screen-reading software, maybe at the library, or another public place.

Finally, the system is live at `http://router-lab.ia.udec.cl:3302/Dropbox/memoria/svn/math/reader/reader.php`

Acknowledgments

This work was supported by an IBM UIMA Innovation Award and a Proyecto de Insercion Postdoctoral PSD 57 del Programa Bicentenario de Ciencia y Tecnologia, CONICYT, Chile, to L. Ferres, and to an Corfo-Innova Bio Bio project for dissertation research to José Fuentes.

For undying, committed kids: find other new directions,
enable creative, yearning types.
– Lao Fu

6. REFERENCES

[1] I. Abu Doush and E. Pontelli. Building a programmable architecture for non-visual navigation of mathematics: Using rules for guiding presentation and switching between modalities. In *UAHCI '09: Proceedings of the 5th International Conference on Universal Access in Human-Computer Interaction. Part III*, pages 3–13, Berlin, Heidelberg, 2009. Springer-Verlag.

[2] W. Bernareggi, C. Jessel, N. Encelle, B. Schweikhardt, and M. Gut. LAMBDA: A European system to access mathematics with Braille and audio synthesis. In K. Miesenberger, J. Klaus, W. Zagler, and A. Karshmer, editors, *Computers Helping People with Special Needs*, volume 4061 of *Lecture Notes in Computer Science*, pages 1223–1230. Springer Berlin / Heidelberg, 2006.

[3] K. V. Deemter, E. Krahmer, and M. Theune. Real versus template-based natural language generation: A false opposition? *Comput. Linguist.*, 31(1):15–24, 2005.

[4] L. Ferres, G. Lindgaard, and L. Sumegi. Evaluating a tool for improving accessibility to charts and graphs. In *Proceedings of the 12th international ACM SIGACCESS conference on Computers and accessibility*, ASSETS '10, pages 83–90, New York, NY, USA, 2010. ACM.

[5] L. Ferres, A. Parush, S. Roberts, and G. Lindgaard. Helping people with visual impairments gain access to graphical information through natural language: The igraph system. In *Proceedings of the 10thICCHP*, Lecture Notes in Computer Science. Springer–Verlag, 2006.

[6] L. Ferres, P. Verkhogliad, G. Lindgaard, L. Boucher, A. Chretien, and M. Lachance. Improving accessibility to statistical graphs: the igraph-lite system. In *Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility*, Assets '07, pages 67–74, New York, NY, USA, 2007. ACM.

[7] Ministerio de Planificación. Encuesta CASEN: Discapadidad. `http://www.mideplan.cl/casen/publicaciones/2006/Resultados_Discapacidad_Casen_2006.pdf`, 2006.

[8] Ministerio de Planificación. Encuesta de caracterización socioeconómica nacional (CASEN). `http://www.mideplan.cl/casen`, 2010.

[9] A. Nemeth. *The Nemeth Braille code for mathematics and science notation: 1972 revision*. Produced in Braille for the Library of Congress, National Library Service for the Blind and Physically Handicapped by the American Printing House for the Blind, 1972.

[10] T. V. Raman. *Audio System For Technical Readings*. PhD thesis, Cornell University, 1994.

[11] H. Reddy and G. Gupta. Dynamic aural browsing of mathml documents with voicexml. In *Human-computer interaction*. Lawrence Erlbaum and Associates, 2005.

[12] E. Reiter. NLG vs. templates. In *Proceedings of the 5th European Workshop in Natural Language Generation*, pages 95–105, Leiden, NL, May 1995.

[13] N. Soiffer. Mathplayer: web-based math accessibility. In *In Assets 05: Proceedings of the 7th international ACM SIGACCESS conference on Computers and accessibility*, pages 204–205. ACM Press, 2005.

[14] S. Sripada and F. Gao. Summarizing dive computer data: A case study in integrating textual and graphical presentations of numerical data. In *Proceedings of Workshop on Multimodal Output Generation*, volume CTIT Proceedings of the Workshop on Multimodal Output Generation, pages 149–157, 2007.

[15] P. Stanley and A. Karshmer. Translating MathML into Nemeth Braille Code. In *International Conference on Computers Helping People with Special Needs*, pages 1175–1182, 2006.

[16] G. Wilcock. Pipelines, templates and transformations: XML for natural language generation. In *Proceedings of the 1st NLP and XML Workshop*, pages 1–8, Tokyo, Japan, 2001.

[15] `http://www.openmath.org/`

[16] `http://webanywhere.cs.washington.edu/`