Variable Shape Logicographic Symbols

KOJI NAKAGAWA

RISC-Linz, Johannes Kepler University, A-4040 Linz, Austria
E-mail: `koji.nakagawa@risc.uni-linz.ac.at`

Abstract

The idea of logicographic symbols is to dispatch graphical drawings to predicate or function constants. The drawings symbolize the intuition behind the notion of the constants. Without losing its rigor logicographic symbols can be used in the formal statements whose readability is remarkably enhanced.

In this paper we propose new type of logicographic symbols, called 'variable shape', which change their shape depending on the arguments whereas others, called 'fixed shape', do not change their shape. Also the design principles which should be took into consideration are discussed.

KEYWORDS: inventing new notation, intuitive proof presentation, user interface for integrated mathematical systems

1. Introduction

The computer-support for doing mathematics has significantly increased in the past few decades. The fact can be seen by the successful achievements of several existing computer algebra systems and theorem proving systems. In these systems mathematical knowledge is described in more formal and precise ways than those of mathematical books so that it can be processed in computers. However the problem is that it is often very hard to understand the intuition behind the formal description in such systems. Traditionally in order to help the understanding, we usually make some pictures or drawings. We definitely need tools to reconcile these two issues, formal rigor and intuitive understanding helped by drawings.

As a solution of this problem Buchberger proposed a new idea of logicographic symbols in a paper [Buc2000]. The main idea of logicographic symbols is to dispatch graphical drawings, which have slots for arguments, to predicate or function constants. The

*) Work was supported by the Project F1302 of the Austrian Science Foundation (FWF).

graphical drawings symbolize the intuition behind the notion. These graphically represented drawings can be treated as meaningful objects. Namely it can be used in formal statements and treated completely same as the traditional formal statements. Additionally these can be not only visible but also manipulatable, e.g by changing arguments by modifying the places called slots.

In order to show the feasibility of logicographic symbols we implemented the tool on the top of the Theorema system which is an integrated mathematical environment for proving, computing, and solving [BDK+1997, BDJ+2000]. So far we have shown several examples of logicographic symbols in several conferences [Buc2000, Buc2001, NB2001a, NB2001b] and my Ph.D. thesis[Nak2002].

The purpose of this paper is to propose a new type of logicographic symbols called 'variable shape' which change their shapes depending on the argument terms. On the contrary the existing type of logicographic symbols are called 'fixed shape' because their shapes are independent of the argument terms.

At first we review the 'fixed shape' logicographic symbols, and secondly see the idea of 'variable shape' logicographic symbols by some examples. Finally we discuss some design principles which should be took into consideration.

2. Fixed Shape Logicographic Symbols

2.1 Example: Limit

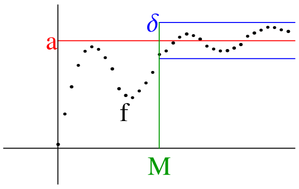
In Theorema the definition of 'sequence f comes closer to a than ϵ from N on' can be described by the following declaration labeled by "4-ary limit" where 'any[f,a,ϵ,N]' indicates that 'f,a,ϵ,N' are variables.

$$\textbf{Definition}\left[\text{"4−ary limit"}, \text{any}[f, a, \epsilon, N], \text{limit}[f, a, \epsilon, N] :\Longleftrightarrow \underset{\substack{n\\ n\geq N}}{\forall} |f[n]-a| < \epsilon\right]$$

Using textual constants for denoting functions and predicates, it is sometimes hard to understand the intuition behind the formalized statements. Proofs of propositions based on this notion will of course rely exclusively on the formal definition. On the other hand a drawing conveying the intuitive idea behind this notion will greatly help in understanding propositions and proofs about the notion. In [Buc2000] Buchberger introduced a logicographic symbol for the predicate constant 'limit' of the formula 'limit[f,a,δ,M]' by the following declaration:

LogicographicNotation["4−ary limit", any[f, a, δ, M],

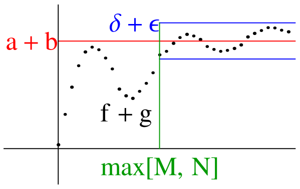
limit[f, a, δ, M] (f " stays closer to " a " than " δ " from " M)

⇋ 

];

In the above declaration, the entire drawing with four *slots* for the four possible arguments constitutes the new symbol. The label "4-ary limit" can be used to refer to this logicographic definition afterwards. The expression 'any[f, a, δ, N]' means that 'f, a, δ, N' are slots in the declaration. The annotation '(f " stays closer to " a " than " δ " from " M)' indicates a suggestion for reading the formula.

After the activation of the declaration, formulae with 4 arguments predicate constant 'limit' are shown by the right-hand of the declaration in which all slots are replaced by the arguments. For example, 'limit[f+g, a+b, $\delta+\epsilon$, max[M,N]]' is shown as follows:



Formulae represented with logicographical symbols can be manipulated just like any other logical formulae of Theorema. Namely, they can be evaluated and their slots can be modified by selecting places of the variable slots and typing the terms.

2.2 Example: Merge Sort

Figure 1 shows a theory for showing the correct of merge sort. The expressions '$\langle\rangle$', '$\langle x, \overline{X}\rangle$', '$x \smile X$', '$X \asymp Y$' stand for 'empty tuple', 'a tuple with the first element x and a finite sequence '$\overline{X}$' of elements', 'tuple X with element x prepended', 'concatenation of X and Y', respectively. And 'stmg', 'mg', 'istv', 'ist', 'ipm', 'lsp', 'rsp' stand for 'sorted by merging', 'merged', 'is sorted version of', 'is sorted', 'is permuted version of', 'left split', and 'right split', respectively.

Algorithm["stmg", any[X],

$$\mathrm{stmg}[X] := \begin{cases} X & \Leftarrow \quad |X| \le 1 \\ \mathrm{mg}[& \Leftarrow \quad \text{otherwise} \\ \quad \mathrm{stmg}[\mathrm{lsp}[X]], & \\ \quad \mathrm{stmg}[\mathrm{rsp}[X]]] & \end{cases} \quad];$$

Algorithm["mg" , any[X, Y, a, b, $\overline{x}$, $\overline{y}$],

$$\begin{array}{l} \mathrm{mg}[\langle\rangle, Y] := Y \\ \mathrm{mg}[X, \langle\rangle] := X \\ \mathrm{mg}[\langle a, \overline{x}\rangle, \langle b, \overline{y}\rangle] := \\ \quad \begin{cases} a \smile \mathrm{mg}[\langle\overline{x}\rangle, \langle b, \overline{y}\rangle] & \Leftarrow \quad a \ge b \\ b \smile \mathrm{mg}[\langle a, \overline{x}\rangle, \langle\overline{y}\rangle] & \Leftarrow \quad \text{otherwise} \end{cases} \end{array} \quad];$$

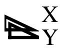
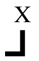
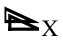
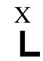
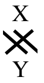
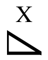
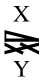
Definition["istv", any[X, Y],
istv[X, Y] $\Longleftrightarrow$ (ist[X] $\wedge$ ipm[X, Y])];

Lemma["mg" , any[A, B],
(ist[A] $\wedge$ ist[B]) $\Longrightarrow$ ist[mg[A, B]]];

Lemma["mg2", any[A, B],
ipm[mg[A, B], A $\asymp$ B]];

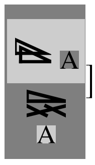
Figure 1: Formalized Merge-Sort Theory

With the several infix notations and the case notation of Theorema, the declaration is much more comprehensive than other programming languages. However no notations are dispatched for the newly introduced constants 'stmg', 'mg', etc. We now introduce logicographic symbols for the newly introduced constants. With the facility of logicographic symbols the user has complete freedom in designing new symbols for the various notions. The following may be a possible choice:

	mg[X, Y]	the result of merging two tuples X and Y		lsp[X]	left split of X
	stmg[X]	the result of sorting X by merging		rsp[X]	right split of X
	ipm[X, Y]	X is a permuted version of Y		ist[X]	X is sorted
	istv[X, Y]	X is a sorted version of Y			

With these logicographic symbols, the above knowledge base can be described in the way shown in Figure 2. Here the expressions are represented in a nested 2-dimensional syntax with dark gray and light gray coloring for clarifying the syntactical structure.

Then the correctness of merge sort can be formalized as follows:

Proposition["mgs" (∗ correctness of merge−sort ∗), any[A], ]

The expected Theorema proof can be found in [NB2001a, Nak2002].

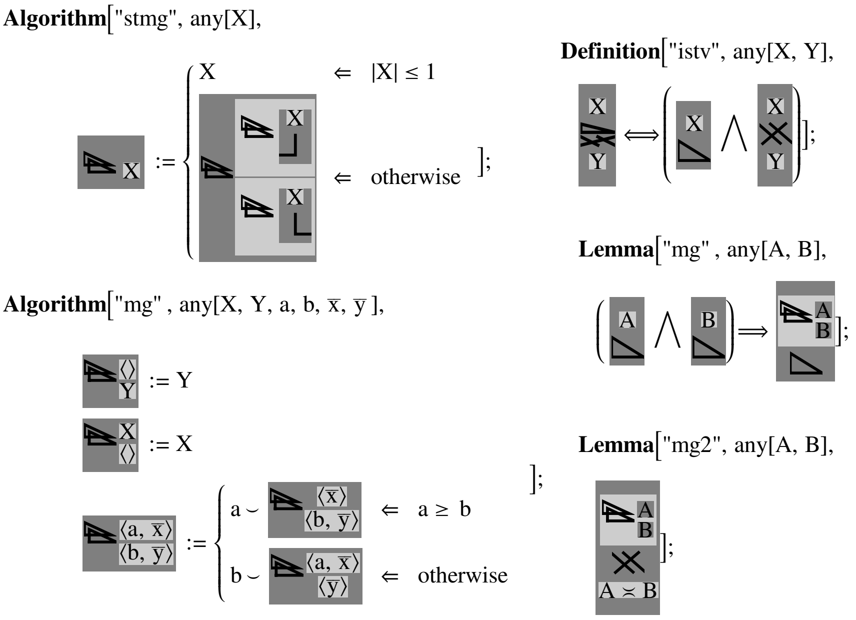


Figure 2: Formalized Merge-Sort Theory with Logicographic Symbols

2.3 Example: Notation of Function in Set Theory

In [Buc2001] Buchberger proposed a technique to compose a new logicographic symbol from existing logicographic symbols by using the example of the notion of function in set theory. Here are the definitions of the notion of function and their logicographic representation:

Definitions["relations", any[A, r, B],

$\text{isrel}[A, r, B] \Longleftrightarrow (r \subseteq A \times B)$

$\text{isltot}[A, r, B] \Longleftrightarrow \underset{a\in A}{\forall}\ \underset{b\in B}{\exists}\ \langle a, b\rangle \in r$

$\text{isrtot}[A, r, B] \Longleftrightarrow \underset{b\in B}{\forall}\ \underset{a\in A}{\exists}\ \langle a, b\rangle \in r$

$\text{isrfun}[A, r, B] \Longleftrightarrow \underset{b1\in B, b2\in B}{\forall}\ \underset{a\in A}{\forall}\ ((\langle a, b1\rangle \in r \land \langle a, b2\rangle \in r \Longrightarrow b1 == b2)$

$\text{islfun}[A, r, B] \Longleftrightarrow \underset{a1\in A, a2\in A}{\forall}\ \underset{b\in B}{\forall}\ ((\langle a1, b\rangle \in r \land \langle a2, b\rangle \in r \Longrightarrow a1 == a2)$

]

LogicographicNotation["relations", any[A, r, B],

isrel[A, r, B] (r "is a relation between " A "and" B) ⇋ $A \overset{r}{\text{———}} B$

isltot[A, r, B] (r "is left total on " A "and" B) ⇋ $A \mid\ ^{r}\ \ B$

isrtot[A, r, B] (r "is right total on " A "and" B) ⇋ $A\ \ ^{r}\ \mid B$

isrfun[A, r, B] (r "is right functional on " A "and" B) ⇋ $A\ \ ^{r}\!> B$

islfun[A, r, B] (r "is left functional on " A "and" B) ⇋ $A <^{r}\ \ B$

]

The notion of being a function can now be expressed by the formula 'isrel[A,r,B]Λisrfun[-A,r,B]' and with the logicographic declaration it is represented as '$A \overset{r}{\text{———}} B \;\Lambda\; A \overset{r}{\;>} B$'. However, it is natural to represent this formula in more compact form '$A \overset{r}{\longrightarrow} B$'.

In order to introduce this representation, we could have an extra 'LogicographicNotation' declaration and its definition. However in this way, in order to introduce all such combined symbols, we would have to introduce 2^5–6(=26) additional logicographic symbols. Instead, at first we introduce a facility to combine existing logicographic symbols by introducing a syntactic construct '∧' (wedge) which is different from the logical construct 'Λ' (and), and second we dispatch appropriate logicographic symbols for it.

For example, if we have the following expression '(isrel∧isrfun)[A,r,B]', then the logicographic representation of the expression are composed from the representation of 'isrel' and 'isfun' and becomes '$A \overset{r}{\longrightarrow} B$'.

Introducing a composing logicographic symbol, e.g. 'isrel∧isrfun', means introducing a new predicate or function constant whose logicographic representation can be automatically composed by combining the logicographic representations of its constituents. Thus, in a Theorema session, these expressions can be treated as predicate or function constants. In fact, the meaning could be defined by just adding definitions of the following kind to the Theorema knowledge base:

Definition["functionality", any[A, r, B],
(isrel ∧ isrfun)[A, r, B] ⟺ isrel[A, r, B] Λ isrfun[A, r, B]]

or alternatively Theorema provers expand the formulae to the appropriate one as the need arises.

Additionally we introduce a new convention. When argument slots in a logicographic symbol are left out, they are considered as quantified argument variables. For example, if the first and third slots are omitted from '$A \overset{r}{\longrightarrow} B$', we will have the representation '$\overset{r}{\longrightarrow}$' which is considered as '$\underset{A}{\exists}\,\underset{B}{\exists}\; A \overset{r}{\longrightarrow} B$', namely '$\underset{A}{\exists}\,\underset{B}{\exists}$ (isrel ∧ isrfun)[A, r, B]'.

Under this conventions the theorem states that 'the composition of two bijective functions is bijective' can be described as follows:

Theorem["Composition of Bijective Functions", any[f, g],
$$\left(\overset{f}{\vdash\!\longleftrightarrow\!\dashv} \;\Lambda\; \overset{g}{\vdash\!\longleftrightarrow\!\dashv}\right) \Rightarrow \overset{f\circ g}{\vdash\!\longleftrightarrow\!\dashv}\;]$$

With 3 lemmata the proof is automatically produced by Theorema as follows:

Prove:

(Theorem (Composition of Bijective Functions))

$\underset{f,g}{\forall}\Big(\overset{f}{\vdash\!\!\longleftrightarrow\!\!\dashv} \bigwedge \overset{g}{\vdash\!\!\longleftrightarrow\!\!\dashv} \Rightarrow \overset{f\circ g}{\vdash\!\!\longleftrightarrow\!\!\dashv} \Big),$

under the assumptions:

(Lemma (Composition of Injective Functions)) $\underset{f,g}{\forall}\Big(\overset{f}{\longleftrightarrow} \bigwedge \overset{g}{\longleftrightarrow} \Rightarrow \overset{f\circ g}{\longleftrightarrow} \Big),$

(Lemma (Existence of Bijective)) $\underset{f}{\forall}\Big(\overset{f}{\longleftrightarrow} \Rightarrow |f \overset{f}{\vdash\!\!\longleftrightarrow\!\!\dashv} f| \Big),$

(Lemma (Bijective is Injective)) $\underset{f}{\forall}\Big(\overset{f}{\vdash\!\!\longleftrightarrow\!\!\dashv} \Rightarrow \overset{f}{\longleftrightarrow} \Big),$

For proving (Theorem (Composition of Bijective Functions)) we take all variables arbitrary but fixed and prove:

(1) $\overset{f_0}{\vdash\!\!\longleftrightarrow\!\!\dashv} \bigwedge \overset{g_0}{\vdash\!\!\longleftrightarrow\!\!\dashv} \Rightarrow \overset{f_0\circ g_0}{\vdash\!\!\longleftrightarrow\!\!\dashv}.$

We prove (1) by the deduction rule.

We assume

(2) $\overset{f_0}{\vdash\!\!\longleftrightarrow\!\!\dashv} \bigwedge \overset{g_0}{\vdash\!\!\longleftrightarrow\!\!\dashv}$

and show

(3) $\overset{f_0\circ g_0}{\vdash\!\!\longleftrightarrow\!\!\dashv}.$

From (2.2), by (Lemma (Bijective is Injective)), we obtain:

(10) $\overset{g_0}{\longleftrightarrow}.$

From (2.1), by (Lemma (Bijective is Injective)), we obtain:

(9) $\overset{f_0}{\longleftrightarrow}.$

From (9) and (10), by (Lemma (Composition of Injective Functions)), we obtain:

(16) $\overset{f_0\circ g_0}{\longleftrightarrow}.$

From (16), by (Lemma (Existence of Bijective)), we obtain:

(26) $|f_0\circ g_0 \overset{f_0\circ g_0}{\vdash\!\!\longleftrightarrow\!\!\dashv} f_0\circ g_0|.$

Formula (3) is proved because (26) is an witness for it.

□

The flow of the produced proof can be seen more legibly by using the following diagrammatic proof presentation. [1] We assume the left hand side and prove the right hand side. [2] In order to prove the goal, we have to find appropriate A and B. [3,4] *Optically* we can grasp the facts by extraction. [5,6] By the lemmata. [7] We found appropriate values for A and B. Therefore the theorem is proved. Producing such diagrammatic proof presentation automatically is an interesting future work.

$$\begin{tikzcd}
\overset{\mathrm{f}}{\vdash\!\!\longleftrightarrow\!\!\dashv} \arrow[d, "{[4]}"] & \bigwedge & \overset{\mathrm{g}}{\vdash\!\!\longleftrightarrow\!\!\dashv} \arrow[d, "{[3]}"] \arrow[rrr, "\text{deduction}", "{[1]}"'] & & & \overset{\mathrm{f}\circ\mathrm{g}}{\vdash\!\!\longleftrightarrow\!\!\dashv} \arrow[d, "{[2]}"] & \\
\overset{\mathrm{f}}{\longleftrightarrow} \arrow[dr] & & \overset{\mathrm{g}}{\longleftrightarrow} \arrow[dl] & & \mathrm{A}^{?} & \overset{\mathrm{f}\circ\mathrm{g}}{\vdash\!\!\longleftrightarrow\!\!\dashv} & \mathrm{B}^{?} \\
& \underset{[5]}{\overset{\mathrm{f}\circ\mathrm{g}}{\longleftrightarrow}} \arrow[rrr, "{[6]}"] & & & |\,\mathrm{f}\circ\mathrm{g} \arrow[u, mapsto] & \overset{\mathrm{f}\circ\mathrm{g}}{\vdash\!\!\longleftrightarrow\!\!\dashv} \arrow[u, phantom, "{[7]}"] & \mathrm{f}\circ\mathrm{g}\,| \arrow[u, mapsto]
\end{tikzcd}$$

3. Variable Shape Logicographic Symbols

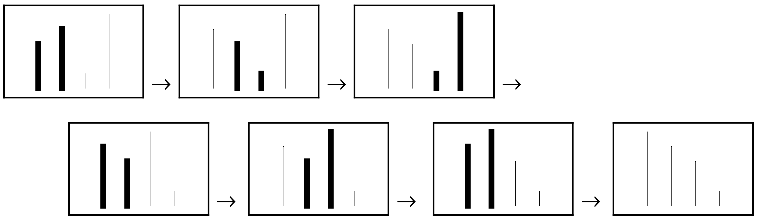
3.1 Example: List Representation

In the previous section we saw some logicographic symbols whose sizes may change, but shape did not change. Namely the shape depends on only the predicate or function constant, but does not depend on the arguments. Here we introduce some logicographic symbols which change their shape depending on the arguments, namely 'variable shape' logicographic symbols.

In mathematics, for heuristic and pedagogic reasons, one often introduces or illustrates abstract notions by accompanying concrete examples. For example, when explaining the bubble-sort algorithm, one would illustrate the effect of the algorithm by showing its trace in concrete examples using lists of numbers. Preferably, one would illustrate the effect by using lists of strokes with varying length so that the effect is more easily visible. For example, when we have the following trace:

⟨**3**, **4**, 1, 5⟩ → ⟨4, **3**, **1**, 5⟩ → ⟨4, 3, **1**, **5**⟩ →
⟨**4**, **3**, 5, 1⟩ → ⟨4, **3**, **5**, 1⟩ → ⟨**4**, **5**, 3, 1⟩ → ⟨5, 4, 3, 1⟩

the corresponding visualized trace is more illustrative than the trace above:



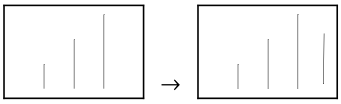
Here lengths of vertical lines vary depending on other elements such that the largest element fits into the box. Also widths between lines vary depending on the number of elements. Note that this is only possible for terms whose ingredient values are known. For terms whose subterms contain variable this is of course not possible.

Variable shape logicographic symbols can not be declared as easily LogicographicNotation declaration used in the examples of the fixed shape logicographic symbols. Instead there should be a mechanism to declare an algorithm to construct the visual representation. For the moment, this functionality has not yet been implemented, but hard-corded in Mathematica.

• Interaction

So far what we could interact with logicographic symbols was to fill in or change the expressions of slots. For this logicographic symbol, different type of interactions is possible. The possible interactions are adding, deleting, arranging, enlarging or shortening bars by mouse operations. These interactions change the elements of lists, e.g. adding an element to the list.

For example, adding an element '2' to the end of the list ⟨1,2,3⟩ can be achieved by drawing a bar into the logicographic symbol:



Note that the system automatically recognizes the size of the element from the vertical length of the bar.

• Tricky Order

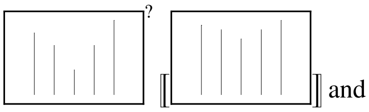
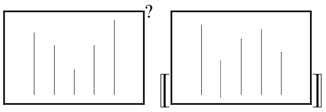
We now consider another usage, which is more general and more tricky: Consider the predicate

$$\textbf{Definition}\Big[\text{"Tricky Order"}, \text{any}[\text{standard}, A], \\ \text{is–ordered}[\text{standard}, A] :\Longleftrightarrow \\ \underset{i=2,\ldots,|\text{standard}|-1}{\forall} \; \underset{j=1,\ldots,i-1}{\forall} (\text{standard}_i \geq \text{standard}_j \Longleftrightarrow A_i \geq A_j)\Big]$$

This predicate yields "True" iff the given list 'A' is ordered exactly in the way as specified by the ordering of the list 'standard'. Now, it may be interesting to introduce an extra logicographic symbol for each concrete value of the parameter 'standard'. In other words, the logicographic representation for the predicate 'is-ordered' would depend on the value of the first argument 'standard'. A possible choice would be, for example:

$$\textbf{LogicographicNotation}[\text{"4−ary limit"}, \text{any}[A,\ \text{standard}], \\ \text{is–ordered}[\text{standard}, A]\ (A \text{ " is ordered by" standard}) \\ \leftrightharpoons \text{standard}^{?}\, [\![A]\!]];$$

Then

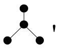
 and 

would yield "True" and "False" respectively.

3.2 Example: Graph Representation

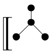
One of natural applications of variable shape logicographic symbols is the graph theory. In the graph theory a graph can be defined as a pair of a set of vertices and a set of edges. For example, this is a graph which has 4 vertices with 3 edges

graph[{v1, v2, v3, v4}, {{v1, v2}, {v1, v3}, {v1, v4}}] .

Then the natural logicographic representation of the graph is ''.

Obviously for humans it is much easier to understand formalized statements of the graph theory by this graphical representation than the formula representation. The following example, taken from a book [Gib1985], shows the computation sequence of chromatic polynomial of a graph.

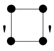
By 'chromatic–polynomial[G,k]', represented logicographically by $[\![G]\!]_k$, we denote the number of ways of vertex-coloring the graph 'G' with 'k' colors such that verities of 'G' should not have the same color if they are adjacent by the edges of 'G'. For example,

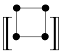
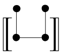
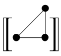
$[\![$  $]\!]_k = k\,(k-1)^3$

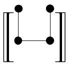
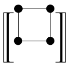
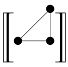
because the vertex in the center can be colored in k different ways. The remaining vertices can then be colored in (k–1) ways. Now here is a useful theorem to compute chromatic polynomials:

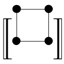
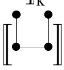
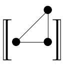
Theorem["7.10 in the book[Gib1985]", any[G, e],
edge[G, e] $\Longrightarrow [\![G]\!]_k = [\![G-e]\!]_k - [\![G\circ e]\!]_k$]

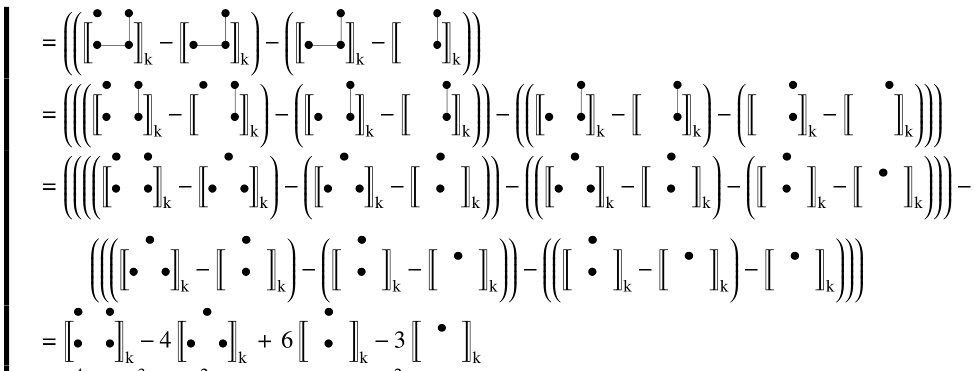
where 'edge[G,e]' means 'e' is an edge of 'G', 'G–e' is derived from 'G' by deleting the edge 'e', and 'G ∘ e' is obtained from G by contracting the edge 'e'. Intuitively contraction of an edge means making the vertices of the edge into one vertex.

We see the idea of the proof by an example. Let 'G' be '' and 'e' be ' ', by the theorem the following fact holds:

$[\![$  $]\!]_k = [\![$  $]\!]_k - [\![$  $]\!]_k$

Here $[\![$  $]\!]_k$ contains the all cases of $[\![$  $]\!]_k$ except the cases where the two upper vertices are the same color. And the number of the exceptional cases is exactly $[\![$  $]\!]_k$. Repeated application of this theorem will eventually reach to the combination of chromatic polynomials with no edges which can be easily computed.

$[\![$  $]\!]_k$
$= [\![$  $]\!]_k - [\![$  $]\!]_k$



$$= k^4 - 4k^3 + 6k^2 - 3k = k(k-1)(k^2 - 3k + 3)$$

With this logicographic representation we will be able to show formal statements of the graph theory more intuitively, and also use the logicographic representation as formally manipulatable expressions, e.g. for computation.

3.3 Example: Geometrical Judgement

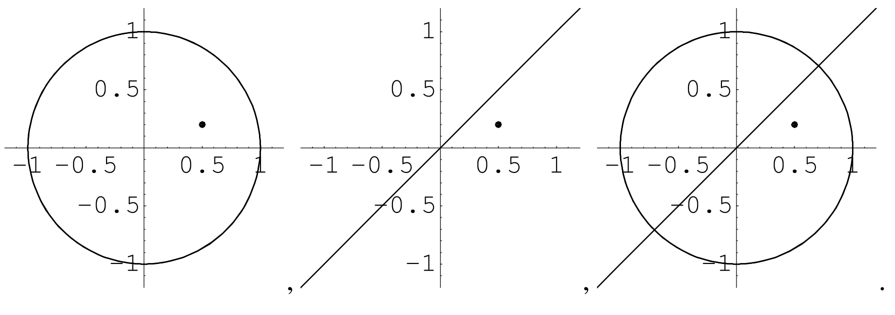
It is a natural idea to use a logicographic symbol for algebraic inequalities as geometrical representation. As an application, the judgement of the region inclusion can be converted into the judgement whether an inequality holds or not. For example, suppose we have the following definitions:

Definitions["in a region", any[G, e],
in–circle[x, y] : $\Longleftrightarrow x^2 + y^2 < 1$
under–line[x, y] : $\Longleftrightarrow y < x$]

Then the formulae

in–circle[0.5, 0.2], under–line[0.5, 0.2], (in–circle ∧ under–line)[0.5, 0.2]

should be represented logicographically in the following ways respectively:



The important observation is '(in–circle∧under–line)[0.5,0.2]' which can be composed from the representations of 'in–circle' and 'under–line' by using the technique explained in Section 2.3. By seeing the picture, one can reason immediately the truth value of formulae without computing algebraic inequalities. (Of course when a point is very close to curves or lines, one can not judge by seeing.)

This kind of reasoning are related to the topic so called 'logical reasoning with diagrams'. One famous example is the Venn diagram by which one can reason some facts with the graphical operations. Shin[Shi1995] and Hammer[Ham1995] improved the Venn diagram and elaborated it in such ways that certain theorems are proved diagrammatically. They also gave soundness and completeness proofs. This is a counter argument against the prejudice that using diagrams is unreliable and that they should not be used in formal proofs. If it is well constructed, one does not lose formality at all. In more general settings the discussions of the properties which these reasoning systems should satisfy can be found in [BS1995, Shi2001].

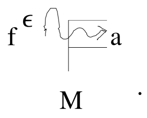
4. Designing Logicographic Symbols

Traditionally, one of the important rules for the invention of notation has always been that it should be easy and fast to write the notation on paper or blackboard. However this rule is not any more so important because inputting becomes faster and easier also for quite involved symbols if we use computers. With this new tool, the users of mathematical systems like Theorema now have a potentially unlimited collection of mathematical symbols at their disposition for expressing their ideas. The design of the logicographic symbol is completely free. However, there are some pedagogic, psychological and design principles which should be followed.

- **Should be as simple as possible and as complicated as necessary**

Most importantly, logicographic symbols should be designed as simple as possible and as complicated as necessary: If a symbol becomes too complicated, it is, again, hard to grasp the essential features of the notion quickly. If a symbol is too simple, it may omit essential features that distinguish the notion from other, similar, notions.

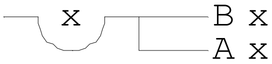
However, the exact design of a logicographic symbol may depend on the pedagogical situation of the addressee in a formal mathematical text: If the notion to be represented by a logicographic symbol is completely new to the addressee, it may be preferable to design a more involved logicographic symbol like the logicographic symbol for 'limit' in Section 2.1. If the notion becomes more and more familiar, an alternative, more concise version of the symbol may be introduced. In fact, it is perfectly possible to attach two or more logicographic symbols with one notion. For example, another possibility could be



• Should reflect the concept behind

Secondly the logicographic symbols should reflect the concept behind. For example, we should not use 'ascendant triangle: $\overset{\text{X}}{\text{◿}}$' for the notion of the predicate checking whether a list is in descendant order or not.

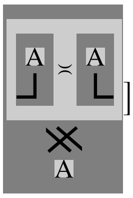
In 1879, Frege introduced 2-dimensional notation for logical formulae [Fre1967]. For example, the formula '$\forall_{x}$ (A[x]$\Longrightarrow$B[x])' is described as follows:



Frege's notation, however, shows only the syntactical structure of formula, whereas logicographic symbols try to convey the intuitive semantics behind the logical constants. Obviously we could implement logicographic symbols for such notation. However the Frege's notation does not convey the intuitive semantics.

• Should use vertical direction and colors

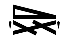
Usually infix notations are located horizontally e.g 'lsp[A]≍rsp[A]'. On the contrary, in the example of 'merge sort' some logicographic symbols are located vertically, too. For example, an axiom 'splitting and concatenating gives a permuted version of the original' can be described with logicographic symbols as follows:

Axiom["csp" , any[A], ]

It uses vertical direction for '✕' which makes the structure clear with the help of colors and enhances the readability. If we do not use the vertical direction and colors, it would be represented as '(┛ [A] ≍ ┗[A])✕A' which does not make us understood immediately. Of course, the above logicographic representation consumes a lot of 2-dimentional space whereas traditional representation uses less 1-dimentional space. However, as mentioned, it is not a problem at all if we use computers, and always we can switch the logicographic symbol depending on the situation.

- **Composite symbols have the meaning of all continents**

The tool of composing representations helps for the notational conciseness. In some sense composed representation have the all properties of its constituents.

For example, in the 'merge sort' theory, the leading picture '' is combined by two existing pictures ' ' and ' ' which indicate that this expression is in some sense have both meanings of constituents as the definition of 'is sorted version'.

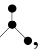
Another example in the proof of 'merge sort', the lemma "Bijective is Injective" is used twice. Note that one can see this fact immediately without referring the lemma because we constructed the logicographic symbols in such a way that the properties represented by a part of a logicographic symbol can be implied from the properties represented by the entire symbol. In this way, one can immediately notice the facts like ' $\overset{f}{\rightarrow\!\mid}$ ', ' $\overset{f}{\mid\!\leftarrow}$ ', ' $\overset{f}{\mid\!-\!\mid}$ ' etc. from ' $\overset{f}{\mid\!\leftrightarrow\!\mid}$ '.

5. Conclusion

After reviewing the examples of fixed shape logicographic symbols, the idea of variable shape logicographic symbols which change their shapes depending on the arguments is proposed by some examples. There are many applications of variable logicographic symbols, e.g. graph theory, geometry, topology, category theory, term rewriting theory etc.

The problem of creating variable shape logicographic symbols is that the drawing and interpreting algorithm should be described on a case basis. Probably there should be a library which consists of some basic functions helping to draw pictures easily and interpreting meanings from drawing components like lines, circles etc.

There are many existing systems which can visualize mathematical objects. However these systems can not use these visualized representation for evaluation. With the tool of logicographic symbols one can freely interchange between visual thinking and logical thinking. This is a distinctive characteristic which can not be seen in other mathematical systems.

In mathematics there are some parts which visual description is suited for and other parts which logical description is suited for. As a tendency, visual description is suited for concrete examples and logical description is suited for describing general properties. For example, let 'P' be a certain property then it's difficult to describe the statement '$\underset{e}{\forall}$ $\Big($edge$\Big[$, e$\Big] \Longrightarrow$ P$\Big[$ $-$ e $\Big]\Big)$' only by visual representation, because 'e' is not concrete yet.

We believe that the tool of logicographic symbols presented in this paper should be integrated into mathematical software systems in future.

References

[BDJ+2000] B. Buchberger, C. Dupre, T. Jebelean, F. Kriftner, K. Nakagawa, D. Vasaru, and W. Windsteiger. The THEOREMA Project: A Progress Report. In *CALCULEMUS 2000 (International Workshop on Systems for Integrated Computation and Deduction)*, St. Andrews, Scotland, August 2000.

[BJK+1997] B. Buchberger, T. Jebelean, F. Kriftner, M. Marin, E. Tomuta, and D. Vasaru. A Survey on the Theorema Project. In W. Kuechlin, editor, *ISSAC'97 (International Symposium on Symbolic and Algebraic Computation)*, pages 384–391, Maui, Hawaii, ACM Press, July 21-23 1997. Also available as RISC technical report 97-15.

[BS1995] J. Barwise and A. Shimojima, Surrogate Reasoning. Cognitive Studies: Bulletin of the Japanese Cognitive Science Society, Vol. 2, No. 4, pp.7-26. 1995.

[Buc2000] B. Buchberger. Logicographic Symbols: Some Examples of Their Use in Formal Proofs. RISC (Research Institute for Symbolic Computation), Johannes Kepler University, Linz, Austria, Feb. 2000. Manuscript.

[Buc2001] B. Buchberger. Logicographic Symbols: A New Feature in Theorema. In Y. Tazawa, editor, *Fourth International Mathematica Symposium (IMS 2001)*, Chiba, Tokyo Denki University. June 2001.

[Fre1967] G. Frege. Begriffsschrift, a Formula Language, Modeled upon that of Arithmetic, for Pure Thought. In *From Frege to Gödel: a Source Book in Mathematical Logic, 1879—1931*, pages 1—82. Harvard University Press, 1967. Original in German in 1879.

[Gib1985] A. Gibbons. Algorithmic Graph Theory. Cambridge University Press, 1985. p. 203.

[Ham1995] E. M. Hammer. *Logic and Visual Information*, CSLI Publications, Stanford, California, 1995.

[NB2001a] K. Nakagawa and B. Buchberger. Presenting Proofs Using Logicographic Symbols. In *Proceedings of the Workshop on Proof Transformation and Presentation*, page 11, Siena, Italy, 2001. http://www.scch.at/research/publications/522/index.html. PTP-01 in IJCAR-2001.

[NB2001b] K. Nakagawa and B. Buchberger. Two Tools for Mathematical Knowledge Management in Theorema. In B. Buchberger and O. Caprotti, editor, *First International Workshop on Mathematical Knowledge Management (MKM 2001)*, Schloss Hagenberg, Austria, RISC. September 2001. http://www.scch.at/research/publications/702/index.html.

[Nak2002] K. Nakagawa. Supporting User-Friendliness in the Mathematical Software System Theroema. Ph.D. thesis of Research Institute for Symbolic Computation, January 2002.

[Shi1995] S. Shin. *The Logical Status of Diagrams*, Cambridge University Press, Cambridge, England, 1995.

[Shi12001] A. Shimojima, A Logical Analysis of Graphical Consistency Proof. Abstracts of the International Conference of Model-Based Reasoning, p. 41, 2001.