

Text-based input formats for mathematical formulas

Peter Jipsen

Chapman University

December 8, 2006

The problem

How to make computers display and understand e.g.:

$$\sin^{-1} \sqrt{\log_e e} = \frac{\pi}{2}$$

Mathematical notation uses complex 2D positioning

The information has to be **entered** in some form

Converted to an internal representation

Displayed / printed / spoken / archived / searched / ...

The problem

How to make computers display and understand e.g.:

$$\sin^{-1} \sqrt{\log_e e} = \frac{\pi}{2}$$

Mathematical notation uses complex 2D positioning

The information has to be **entered** in some form

Converted to an internal representation

Displayed / printed / spoken / archived / searched / ...

The problem

How to make computers display and understand e.g.:

$$\sin^{-1} \sqrt{\log_e e} = \frac{\pi}{2}$$

Mathematical notation uses complex 2D positioning

The information has to be **entered** in some form

Converted to an internal representation

Displayed / printed / spoken / archived / searched / ...

The problem

How to make computers display and understand e.g.:

$$\sin^{-1} \sqrt{\log_e e} = \frac{\pi}{2}$$

Mathematical notation uses complex 2D positioning

The information has to be **entered** in some form

Converted to an internal representation

Displayed / printed / spoken / archived / searched / ...

The problem

How to make computers display and understand e.g.:

$$\sin^{-1} \sqrt{\log_e e} = \frac{\pi}{2}$$

Mathematical notation uses complex 2D positioning

The information has to be **entered** in some form

Converted to an internal representation

Displayed / printed / spoken / archived / searched / ...

Creating mathematical content

Traditional document: **Handwritten**

Advantages

- versatile
- simple
- fast

Disadvantages

- hard to digitize
- hard to parse
- can't edit or copy/paste easily
- semantics?

Creating mathematical content

Traditional document: **Handwritten**

Advantages

- versatile
- simple
- fast

Disadvantages

- hard to digitize
- hard to parse
- can't edit or copy/paste easily
- semantics?

Creating mathematical content

Traditional document: **Handwritten**

Advantages

- versatile
- simple
- fast

Disadvantages

- hard to digitize
- hard to parse
- can't edit or copy/paste easily
- semantics?

Creating mathematical content

Traditional document: **Handwritten**

Advantages

- versatile
- simple
- fast

Disadvantages

- hard to digitize
- hard to parse
- can't edit or copy/paste easily
- semantics?

Creating mathematical content

Traditional document: **Handwritten**

Advantages

- versatile
- simple
- fast

Disadvantages

- hard to digitize
- hard to parse
- can't edit or copy/paste easily
- semantics?

Creating mathematical content

Traditional document: **Handwritten**

Advantages

- versatile
- simple
- fast

Disadvantages

- hard to digitize
- hard to parse
- can't edit or copy/paste easily
- semantics?

Creating mathematical content

Traditional document: **Handwritten**

Advantages

- versatile
- simple
- fast

Disadvantages

- hard to digitize
- hard to parse
- can't edit or copy/paste easily
- semantics?

Creating mathematical content

Traditional document: **Handwritten**

Advantages

- versatile
- simple
- fast

Disadvantages

- hard to digitize
- hard to parse
- can't edit or copy/paste easily
- semantics?

Creating mathematical content

Traditional document: **Handwritten**

Advantages

- versatile
- simple
- fast

Disadvantages

- hard to digitize
- hard to parse
- can't edit or copy/paste easily
- semantics?

Creating mathematical content

Traditional document: using point and click formula editor

Advantages

- easy to use
- wysiwyg
- captures structure

Disadvantages

- slow
- nonstandard
- difficult to add to existing tools
- display quality?

Creating mathematical content

Traditional document: using point and click formula editor

Advantages

- easy to use
- wysiwyg
- captures structure

Disadvantages

- slow
- nonstandard
- difficult to add to existing tools
- display quality?

Creating mathematical content

Traditional document: using point and click formula editor

Advantages

- easy to use
- wysiwyg
- captures structure

Disadvantages

- slow
- nonstandard
- difficult to add to existing tools
- display quality?

Creating mathematical content

Traditional document: using point and click formula editor

Advantages

- easy to use
- wysiwyg
- captures structure

Disadvantages

- slow
- nonstandard
- difficult to add to existing tools
- display quality?

Creating mathematical content

Traditional document: using point and click formula editor

Advantages

- easy to use
- wysiwyg
- captures structure

Disadvantages

- slow
- nonstandard
- difficult to add to existing tools
- display quality?

Creating mathematical content

Traditional document: using point and click formula editor

Advantages

- easy to use
- wysiwyg
- captures structure

Disadvantages

- slow
- nonstandard
- difficult to add to existing tools
- display quality?

Creating mathematical content

Traditional document: using point and click formula editor

Advantages

- easy to use
- wysiwyg
- captures structure

Disadvantages

- slow
- nonstandard
- difficult to add to existing tools
- display quality?

Creating mathematical content

Traditional document: using point and click formula editor

Advantages

- easy to use
- wysiwyg
- captures structure

Disadvantages

- slow
- nonstandard
- difficult to add to existing tools
- display quality?

Creating mathematical content

Traditional document: using point and click formula editor

Advantages

- easy to use
- wysiwyg
- captures structure

Disadvantages

- slow
- nonstandard
- difficult to add to existing tools
- display quality?

Creating mathematical content

Traditional document: **using a typesetting system**

Advantages

- high quality output
- import/export features for larger systems
- expected by publishers

Disadvantages

- cryptic commands
- tedious textediting/proofreading
- “nonstandard”

Creating mathematical content

Traditional document: **using a typesetting system**

Advantages

- high quality output
- import/export features for larger systems
- expected by publishers

Disadvantages

- cryptic commands
- tedious textediting/proofreading
- “nonstandard”

Creating mathematical content

Traditional document: **using a typesetting system**

Advantages

- high quality output
- import/export features for larger systems
- expected by publishers

Disadvantages

- cryptic commands
- tedious textediting/proofreading
- “nonstandard”

Creating mathematical content

Traditional document: **using a typesetting system**

Advantages

- high quality output
- import/export features for larger systems
- expected by publishers

Disadvantages

- cryptic commands
- tedious textediting/proofreading
- “nonstandard”

Creating mathematical content

Traditional document: **using a typesetting system**

Advantages

- high quality output
- import/export features for larger systems
- expected by publishers

Disadvantages

- cryptic commands
- tedious textediting/proofreading
- “nonstandard”

Creating mathematical content

Traditional document: **using a typesetting system**

Advantages

- high quality output
- import/export features for larger systems
- expected by publishers

Disadvantages

- cryptic commands
- tedious textediting/proofreading
- “nonstandard”

Creating mathematical content

Traditional document: **using a typesetting system**

Advantages

- high quality output
- import/export features for larger systems
- expected by publishers

Disadvantages

- cryptic commands
- tedious textediting/proofreading
- “nonstandard”

Creating mathematical content

Traditional document: **using a typesetting system**

Advantages

- high quality output
- import/export features for larger systems
- expected by publishers

Disadvantages

- cryptic commands
- tedious textediting/proofreading
- “nonstandard”

Displaying mathematical content

Math on webpages

- scan handwritten pages
 - post digital photo of a whiteboard
 - use a tablet PC, post input as picture
 - post a video of a presentation
-
- use a converter to change each formula to gif or png (e.g. LaTeX2HTML, Wikipedia, ...)
 - use HTML/ASCII approximations (hand edit, TtH, TeX2HTML, ...)
 - position fonts with CSS (jsMath, ...)
 - MathML

Displaying mathematical content

Math on webpages

- scan handwritten pages
 - post digital photo of a whiteboard
 - use a tablet PC, post input as picture
 - post a video of a presentation
-
- use a converter to change each formula to gif or png (e.g. LaTeX2HTML, Wikipedia, ...)
 - use HTML/ASCII approximations (hand edit, TtH, TeX2HTML, ...)
 - position fonts with CSS (jsMath, ...)
 - MathML

Displaying mathematical content

Math on webpages

- scan handwritten pages
 - post digital photo of a whiteboard
 - use a tablet PC, post input as picture
 - post a video of a presentation
-
- use a converter to change each formula to gif or png (e.g. LaTeX2HTML, Wikipedia, ...)
 - use HTML/ASCII approximations (hand edit, TtH, TeX2HTML, ...)
 - position fonts with CSS (jsMath, ...)
 - MathML

Displaying mathematical content

Math on webpages

- scan handwritten pages
 - post digital photo of a whiteboard
 - use a tablet PC, post input as picture
 - post a video of a presentation
-
- use a converter to change each formula to gif or png (e.g. LaTeX2HTML, Wikipedia, ...)
 - use HTML/ASCII approximations (hand edit, TtH, TeX2HTML, ...)
 - position fonts with CSS (jsMath, ...)
 - MathML

Displaying mathematical content

Math on webpages

- scan handwritten pages
 - post digital photo of a whiteboard
 - use a tablet PC, post input as picture
 - post a video of a presentation
-
- use a converter to change each formula to gif or png (e.g. LaTeX2HTML, Wikipedia, ...)
 - use HTML/ASCII approximations (hand edit, TtH, TeX2HTML, ...)
 - position fonts with CSS (jsMath, ...)
 - MathML

Displaying mathematical content

Math on webpages

- scan handwritten pages
 - post digital photo of a whiteboard
 - use a tablet PC, post input as picture
 - post a video of a presentation
-
- use a converter to change each formula to gif or png (e.g. LaTeX2HTML, Wikipedia, ...)
 - use HTML/ASCII approximations (hand edit, TtH, TeX2HTML, ...)
 - position fonts with CSS (jsMath, ...)
 - MathML

Displaying mathematical content

Math on webpages

- scan handwritten pages
 - post digital photo of a whiteboard
 - use a tablet PC, post input as picture
 - post a video of a presentation
-
- use a converter to change each formula to gif or png (e.g. LaTeX2HTML, Wikipedia, ...)
 - use HTML/ASCII approximations (hand edit, TtH, TeX2HTML, ...)
 - position fonts with CSS (jsMath, ...)
 - MathML

Displaying mathematical content

Math on webpages

- scan handwritten pages
 - post digital photo of a whiteboard
 - use a tablet PC, post input as picture
 - post a video of a presentation
-
- use a converter to change each formula to gif or png (e.g. LaTeX2HTML, Wikipedia, ...)
 - use HTML/ASCII approximations (hand edit, TtH, TeX2HTML, ...)
 - position fonts with CSS (jsMath, ...)
 - MathML

Displaying mathematical content

Math on webpages

- scan handwritten pages
 - post digital photo of a whiteboard
 - use a tablet PC, post input as picture
 - post a video of a presentation
-
- use a converter to change each formula to gif or png (e.g. LaTeX2HTML, Wikipedia, ...)
 - use HTML/ASCII approximations (hand edit, TtH, TeX2HTML, ...)
 - position fonts with CSS (jsMath, ...)
 - MathML

Displaying mathematical content

Math on webpages

- scan handwritten pages
 - post digital photo of a whiteboard
 - use a tablet PC, post input as picture
 - post a video of a presentation
-
- use a converter to change each formula to gif or png (e.g. LaTeX2HTML, Wikipedia, ...)
 - use HTML/ASCII approximations (hand edit, TtH, TeX2HTML, ...)
 - position fonts with CSS (jsMath, ...)
 - MathML

Text based input

- Keyboards are the most widely used form of character-based input
- Likely to remain true for at least another decade
- Want to communicate math content easily
- Chat, read, edit email replies in a non-proprietary way
- A linear character-based format fits well
- Many different math input syntaxes have been developed for
 - ▶ programming languages: Fortran, APL, Lisp, C, Pascal, Java, ...
 - ▶ scripting languages: JS, Perl, PHP, Python, ...
 - ▶ calculators: TI-83, TI-89, Casio FX, HP, ...
 - ▶ computer algebra systems: Macsyma, Reduce, Mathematica, Maple, Scientific Notebook, SAGE, ...
 - ▶ typesetting systems: troff, TeX/LaTeX, DocBook, ...

Text based input

- Keyboards are the most widely used form of character-based input
- Likely to remain true for at least another decade
- Want to communicate math content easily
- Chat, read, edit email replies in a non-proprietary way
- A linear character-based format fits well
- Many different math input syntaxes have been developed for
 - ▶ programming languages: Fortran, APL, Lisp, C, Pascal, Java, ...
 - ▶ scripting languages: JS, Perl, PHP, Python, ...
 - ▶ calculators: TI-83, TI-89, Casio FX, HP, ...
 - ▶ computer algebra systems: Macsyma, Reduce, Mathematica, Maple, Scientific Notebook, SAGE, ...
 - ▶ typesetting systems: troff, TeX/LaTeX, DocBook, ...

Text based input

- Keyboards are the most widely used form of character-based input
- Likely to remain true for at least another decade
- Want to communicate math content easily
- Chat, read, edit email replies in a non-proprietary way
- A linear character-based format fits well
- Many different math input syntaxes have been developed for
 - ▶ programming languages: Fortran, APL, Lisp, C, Pascal, Java, ...
 - ▶ scripting languages: JS, Perl, PHP, Python, ...
 - ▶ calculators: TI-83, TI-89, Casio FX, HP, ...
 - ▶ computer algebra systems: Macsyma, Reduce, Mathematica, Maple, Scientific Notebook, SAGE, ...
 - ▶ typesetting systems: troff, TeX/LaTeX, DocBook, ...

Text based input

- Keyboards are the most widely used form of character-based input
- Likely to remain true for at least another decade
- Want to communicate math content easily
- Chat, read, edit email replies in a non-proprietary way
- A linear character-based format fits well
- Many different math input syntaxes have been developed for
 - ▶ programming languages: Fortran, APL, Lisp, C, Pascal, Java, ...
 - ▶ scripting languages: JS, Perl, PHP, Python, ...
 - ▶ calculators: TI-83, TI-89, Casio FX, HP, ...
 - ▶ computer algebra systems: Macsyma, Reduce, Mathematica, Maple, Scientific Notebook, SAGE, ...
 - ▶ typesetting systems: troff, TeX/LaTeX, DocBook, ...

Text based input

- Keyboards are the most widely used form of character-based input
- Likely to remain true for at least another decade
- Want to communicate math content easily
- Chat, read, edit email replies in a non-proprietary way
- A linear character-based format fits well
- Many different math input syntaxes have been developed for
 - ▶ programming languages: Fortran, APL, Lisp, C, Pascal, Java, ...
 - ▶ scripting languages: JS, Perl, PHP, Python, ...
 - ▶ calculators: TI-83, TI-89, Casio FX, HP, ...
 - ▶ computer algebra systems: Macsyma, Reduce, Mathematica, Maple, Scientific Notebook, SAGE, ...
 - ▶ typesetting systems: troff, TeX/LaTeX, DocBook, ...

Text based input

- Keyboards are the most widely used form of character-based input
- Likely to remain true for at least another decade
- Want to communicate math content easily
- Chat, read, edit email replies in a non-proprietary way
- A linear character-based format fits well
- Many different math input syntaxes have been developed for
 - ▶ programming languages: Fortran, APL, Lisp, C, Pascal, Java, ...
 - ▶ scripting languages: JS, Perl, PHP, Python, ...
 - ▶ calculators: TI-83, TI-89, Casio FX, HP, ...
 - ▶ computer algebra systems: Macsyma, Reduce, Mathematica, Maple, Scientific Notebook, SAGE, ...
 - ▶ typesetting systems: troff, TeX/LaTeX, DocBook, ...

Text based input

- Keyboards are the most widely used form of character-based input
- Likely to remain true for at least another decade
- Want to communicate math content easily
- Chat, read, edit email replies in a non-proprietary way
- A linear character-based format fits well
- Many different math input syntaxes have been developed for
 - ▶ programming languages: Fortran, APL, Lisp, C, Pascal, Java, ...
 - ▶ scripting languages: JS, Perl, PHP, Python, ...
 - ▶ calculators: TI-83, TI-89, Casio FX, HP, ...
 - ▶ computer algebra systems: Macsyma, Reduce, Mathematica, Maple, Scientific Notebook, SAGE, ...
 - ▶ typesetting systems: troff, TeX/LaTeX, DocBook, ...

Text based input

- Keyboards are the most widely used form of character-based input
- Likely to remain true for at least another decade
- Want to communicate math content easily
- Chat, read, edit email replies in a non-proprietary way
- A linear character-based format fits well
- Many different math input syntaxes have been developed for
 - ▶ programming languages: Fortran, APL, Lisp, C, Pascal, Java, ...
 - ▶ scripting languages: JS, Perl, PHP, Python, ...
 - ▶ calculators: TI-83, TI-89, Casio FX, HP, ...
 - ▶ computer algebra systems: Macsyma, Reduce, Mathematica, Maple, Scientific Notebook, SAGE, ...
 - ▶ typesetting systems: troff, TeX/LaTeX, DocBook, ...

Text based input

- Keyboards are the most widely used form of character-based input
- Likely to remain true for at least another decade
- Want to communicate math content easily
- Chat, read, edit email replies in a non-proprietary way
- A linear character-based format fits well
- Many different math input syntaxes have been developed for
 - ▶ programming languages: Fortran, APL, Lisp, C, Pascal, Java, ...
 - ▶ scripting languages: JS, Perl, PHP, Python, ...
 - ▶ calculators: TI-83, TI-89, Casio FX, HP, ...
 - ▶ computer algebra systems: Macsyma, Reduce, Mathematica, Maple, Scientific Notebook, SAGE, ...
 - ▶ typesetting systems: troff, TeX/LaTeX, DocBook, ...

Text based input

- Keyboards are the most widely used form of character-based input
- Likely to remain true for at least another decade
- Want to communicate math content easily
- Chat, read, edit email replies in a non-proprietary way
- A linear character-based format fits well
- Many different math input syntaxes have been developed for
 - ▶ programming languages: Fortran, APL, Lisp, C, Pascal, Java, ...
 - ▶ scripting languages: JS, Perl, PHP, Python, ...
 - ▶ calculators: TI-83, TI-89, Casio FX, HP, ...
 - ▶ computer algebra systems: Macsyma, Reduce, Mathematica, Maple, Scientific Notebook, SAGE, ...
 - ▶ typesetting systems: troff, TeX/LaTeX, DocBook, ...

Text based input

- Keyboards are the most widely used form of character-based input
- Likely to remain true for at least another decade
- Want to communicate math content easily
- Chat, read, edit email replies in a non-proprietary way
- A linear character-based format fits well
- Many different math input syntaxes have been developed for
 - ▶ programming languages: Fortran, APL, Lisp, C, Pascal, Java, ...
 - ▶ scripting languages: JS, Perl, PHP, Python, ...
 - ▶ calculators: TI-83, TI-89, Casio FX, HP, ...
 - ▶ computer algebra systems: Macsyma, Reduce, Mathematica, Maple, Scientific Notebook, SAGE, ...
 - ▶ typesetting systems: troff, TeX/LaTeX, DocBook, ...

Common features of most linear math notation

- Prefix function notation with infix operations $+$, $-$, $*$, \dots
 - Some precedence of operations is used
 - Parenthesis are used for grouping, override precedence
 - Variable names may consist of several characters
 - Incorrect syntax raises errors
-
- The first three are standard in handwritten formulas
 - The other two are less usual in mathematics

Common features of most linear math notation

- Prefix function notation with infix operations $+$, $-$, $*$, \dots
- Some precedence of operations is used
 - Parenthesis are used for grouping, override precedence
 - Variable names may consist of several characters
 - Incorrect syntax raises errors
- The first three are standard in handwritten formulas
- The other two are less usual in mathematics

Common features of most linear math notation

- Prefix function notation with infix operations $+$, $-$, $*$, \dots
 - Some precedence of operations is used
 - Parenthesis are used for grouping, override precedence
 - Variable names may consist of several characters
 - Incorrect syntax raises errors
-
- The first three are standard in handwritten formulas
 - The other two are less usual in mathematics

Common features of most linear math notation

- Prefix function notation with infix operations $+$, $-$, $*$, \dots
 - Some precedence of operations is used
 - Parenthesis are used for grouping, override precedence
 - Variable names may consist of several characters
 - Incorrect syntax raises errors
-
- The first three are standard in handwritten formulas
 - The other two are less usual in mathematics

Common features of most linear math notation

- Prefix function notation with infix operations $+$, $-$, $*$, \dots
 - Some precedence of operations is used
 - Parenthesis are used for grouping, override precedence
 - Variable names may consist of several characters
 - Incorrect syntax raises errors
-
- The first three are standard in handwritten formulas
 - The other two are less usual in mathematics

Common features of most linear math notation

- Prefix function notation with infix operations $+$, $-$, $*$, \dots
 - Some precedence of operations is used
 - Parenthesis are used for grouping, override precedence
 - Variable names may consist of several characters
 - Incorrect syntax raises errors
-
- The first three are standard in handwritten formulas
 - The other two are less usual in mathematics

Common features of most linear math notation

- Prefix function notation with infix operations $+$, $-$, $*$, \dots
 - Some precedence of operations is used
 - Parenthesis are used for grouping, override precedence
 - Variable names may consist of several characters
 - Incorrect syntax raises errors
-
- The first three are standard in handwritten formulas
 - The other two are less usual in mathematics

Many differences of various input formats

For example: $\sin^{-1} \sqrt{\log_e e} = \frac{\pi}{2}$

Many differences of various input formats

For example: $\sin^{-1} \sqrt{\log_e e} = \frac{\pi}{2}$

- `ArcSin[Sqrt[Log[E]]]==Pi/2`

Many differences of various input formats

For example: $\sin^{-1} \sqrt{\log_e e} = \frac{\pi}{2}$

- `ArcSin[Sqrt[Log[E]]]==Pi/2`
Mathematica

Many differences of various input formats

For example: $\sin^{-1} \sqrt{\log_e e} = \frac{\pi}{2}$

- `ArcSin[Sqrt[Log[E]]]==Pi/2`
Mathematica
- `Math.asin(Math.sqrt(Math.log(Math.E)))==Math.Pi/2`

Many differences of various input formats

For example: $\sin^{-1} \sqrt{\log_e e} = \frac{\pi}{2}$

- `ArcSin[Sqrt[Log[E]]]==Pi/2`
Mathematica
- `Math.asin(Math.sqrt(Math.log(Math.E)))==Math.Pi/2`
JavaScript

Many differences of various input formats

For example: $\sin^{-1} \sqrt{\log_e e} = \frac{\pi}{2}$

- `ArcSin[Sqrt[Log[E]]]==Pi/2`
Mathematica
- `Math.asin(Math.sqrt(Math.log(Math.E)))==Math.Pi/2`
JavaScript
- `\sin^{-1}\sqrt{\log_e e}=\frac{\pi}{2}`

Many differences of various input formats

For example: $\sin^{-1} \sqrt{\log_e e} = \frac{\pi}{2}$

- `ArcSin[Sqrt[Log[E]]]==Pi/2`
Mathematica
- `Math.asin(Math.sqrt(Math.log(Math.E)))==Math.Pi/2`
JavaScript
- `\sin^{-1}\sqrt{\log_e e}=\frac{\pi}{2}`
jsMath

Many differences of various input formats

For example: $\sin^{-1} \sqrt{\log_e e} = \frac{\pi}{2}$

- `ArcSin[Sqrt[Log[E]]]==Pi/2`
Mathematica
- `Math.asin(Math.sqrt(Math.log(Math.E)))==Math.Pi/2`
JavaScript
- `\sin^{-1}\sqrt{\log_e e}=\frac{\pi}{2}`
jsMath or LaTeX

Many differences of various input formats

For example: $\sin^{-1} \sqrt{\log_e e} = \frac{\pi}{2}$

- `ArcSin[Sqrt[Log[E]]]==Pi/2`
Mathematica
- `Math.asin(Math.sqrt(Math.log(Math.E)))==Math.Pi/2`
JavaScript
- `\sin^{-1}\sqrt{\log_e e}=\frac{\pi}{2}`
jsMath or LaTeX
- `<msup><mo>sin</mo><mrow><mo>-</mo><mn>1</mn></mrow>
</msup><msqrt><msub><mo>log</mo><mi>e</mi></msub><mi>e</mi>
</msqrt><mo>=</mo><mfrac><mi>π</mi><mn>2</mn></mfrac>`

Many differences of various input formats

For example: $\sin^{-1} \sqrt{\log_e e} = \frac{\pi}{2}$

- `ArcSin[Sqrt[Log[E]]]==Pi/2`

Mathematica

- `Math.asin(Math.sqrt(Math.log(Math.E)))==Math.Pi/2`

JavaScript

- `\sin^{-1}\sqrt{\log_e e}=\frac{\pi}{2}`

jsMath or LaTeX

- `<msup><mo>sin</mo><mrow><mo>-</mo><mn>1</mn></mrow>`

`</msup><msqrt><msub><mo>log</mo><mi>e</mi></msub><mi>e</mi>`

`</msqrt><mo>=</mo><mfrac><mi>π</mi><mn>2</mn></mfrac>`

Presentation MathML

Many differences of various input formats

For example: $\sin^{-1} \sqrt{\log_e e} = \frac{\pi}{2}$

- `ArcSin[Sqrt[Log[E]]]==Pi/2`

Mathematica

- `Math.asin(Math.sqrt(Math.log(Math.E)))==Math.Pi/2`

JavaScript

- `\sin^{-1}\sqrt{\log_e e}=\frac{\pi}{2}`

jsMath or LaTeX

- `<msup><mo>sin</mo><mrow><mo>-</mo><mn>1</mn></mrow>`

`</msup><msqrt><msub><mo>log</mo><mi>e</mi></msub><mi>e</mi>`

`</msqrt><mo>=</mo><mfrac><mi>π</mi><mn>2</mn></mfrac>`

Presentation MathML

- `sin-1sqrt(loge e)=pi/2`

Many differences of various input formats

For example: $\sin^{-1} \sqrt{\log_e e} = \frac{\pi}{2}$

- `ArcSin[Sqrt[Log[E]]]==Pi/2`

Mathematica

- `Math.asin(Math.sqrt(Math.log(Math.E)))==Math.Pi/2`

JavaScript

- `\sin^{-1}\sqrt{\log_e e}=\frac{\pi}{2}`

jsMath or LaTeX

- `<msup><mo>sin</mo><mrow><mo>-</mo><mn>1</mn></mrow>`

`</msup><msqrt><msub><mo>log</mo><mi>e</mi></msub><mi>e</mi>`

`</msqrt><mo>=</mo><mfrac><mi>π</mi><mn>2</mn></mfrac>`

Presentation MathML

- `sin-1sqrt(loge e)=pi/2`

ASCIIMath

Why use formulas for mathematics?

Formulas are

- used to **precisely specify** concepts in a **compact** and **standard** way
- convenient for **manual** manipulation (replacing equals by equals)
- a “**canonical form**” across diverse areas of math
- a common **language** with mnemonic recognition value
- an **informal standard** for math notation; quite international
- **Typed** math notation **deserves** a similar informal standard

- **LaTeX** is a de facto standard for research publications
- But not widely used in **school** or **undergraduate** math
- **Not** compact or easy to read or type (for non-technical users)

Why use formulas for mathematics?

Formulas are

- used to **precisely specify** concepts in a **compact** and **standard** way
- convenient for **manual** manipulation (replacing equals by equals)
- a “**canonical form**” across diverse areas of math
- a common **language** with mnemonic recognition value
- an **informal standard** for math notation; quite international
- **Typed** math notation **deserves** a similar informal standard

- **LaTeX** is a de facto standard for research publications
- But not widely used in **school** or **undergraduate** math
- **Not** compact or easy to read or type (for non-technical users)

Why use formulas for mathematics?

Formulas are

- used to **precisely specify** concepts in a **compact** and **standard** way
- convenient for **manual** manipulation (replacing equals by equals)
- a “**canonical form**” across diverse areas of math
- a common **language** with mnemonic recognition value
- an **informal standard** for math notation; quite international
- **Typed** math notation **deserves** a similar informal standard

- **LaTeX** is a de facto standard for research publications
- But not widely used in **school** or **undergraduate** math
- **Not** compact or easy to read or type (for non-technical users)

Why use formulas for mathematics?

Formulas are

- used to **precisely specify** concepts in a **compact** and **standard** way
- convenient for **manual** manipulation (replacing equals by equals)
- a “**canonical form**” across diverse areas of math
- a common **language** with mnemonic recognition value
- an **informal standard** for math notation; quite international
- **Typed** math notation **deserves** a similar informal standard

- **LaTeX** is a de facto standard for research publications
- But not widely used in **school** or **undergraduate** math
- **Not** compact or easy to read or type (for non-technical users)

Why use formulas for mathematics?

Formulas are

- used to **precisely specify** concepts in a **compact** and **standard** way
- convenient for **manual** manipulation (replacing equals by equals)
- a “**canonical form**” across diverse areas of math
- a common **language** with mnemonic recognition value
- an **informal standard** for math notation; quite international
- **Typed** math notation **deserves** a similar informal standard

- **LaTeX** is a de facto standard for research publications
- But not widely used in **school** or **undergraduate** math
- **Not** compact or easy to read or type (for non-technical users)

Why use formulas for mathematics?

Formulas are

- used to **precisely specify** concepts in a **compact** and **standard** way
- convenient for **manual** manipulation (replacing equals by equals)
- a “**canonical form**” across diverse areas of math
- a common **language** with mnemonic recognition value
- an **informal standard** for math notation; quite international
- **Typed** math notation **deserves** a similar informal standard

- **LaTeX** is a de facto standard for research publications
- But not widely used in **school** or **undergraduate** math
- **Not** compact or easy to read or type (for non-technical users)

Why use formulas for mathematics?

Formulas are

- used to **precisely specify** concepts in a **compact** and **standard** way
- convenient for **manual** manipulation (replacing equals by equals)
- a “**canonical form**” across diverse areas of math
- a common **language** with mnemonic recognition value
- an **informal standard** for math notation; quite international
- **Typed** math notation **deserves** a similar informal standard

- **LaTeX** is a de facto standard for research publications
- But not widely used in **school** or **undergraduate** math
- **Not** compact or easy to read or type (for non-technical users)

Why use formulas for mathematics?

Formulas are

- used to **precisely specify** concepts in a **compact** and **standard** way
- convenient for **manual** manipulation (replacing equals by equals)
- a “**canonical form**” across diverse areas of math
- a common **language** with mnemonic recognition value
- an **informal standard** for math notation; quite international
- **Typed** math notation **deserves** a similar informal standard

- **LaTeX** is a de facto standard for research publications
- But not widely used in **school** or **undergraduate** math
- **Not** compact or easy to read or type (for non-technical users)

Why use formulas for mathematics?

Formulas are

- used to **precisely specify** concepts in a **compact** and **standard** way
 - convenient for **manual** manipulation (replacing equals by equals)
 - a “**canonical form**” across diverse areas of math
 - a common **language** with mnemonic recognition value
 - an **informal standard** for math notation; quite international
 - **Typed** math notation **deserves** a similar informal standard
-
- **LaTeX** is a de facto standard for research publications
 - But not widely used in **school** or **undergraduate** math
 - **Not** compact or easy to read or type (for non-technical users)

Why use formulas for mathematics?

Formulas are

- used to **precisely specify** concepts in a **compact** and **standard** way
- convenient for **manual** manipulation (replacing equals by equals)
- a “**canonical form**” across diverse areas of math
- a common **language** with mnemonic recognition value
- an **informal standard** for math notation; quite international
- **Typed** math notation **deserves** a similar informal standard

- **LaTeX** is a de facto standard for research publications
- But not widely used in **school** or **undergraduate** math
- **Not** compact or easy to read or type (for non-technical users)

Why use formulas for mathematics?

Formulas are

- used to **precisely specify** concepts in a **compact** and **standard** way
- convenient for **manual** manipulation (replacing equals by equals)
- a “**canonical form**” across diverse areas of math
- a common **language** with mnemonic recognition value
- an **informal standard** for math notation; quite international
- **Typed** math notation **deserves** a similar informal standard

- **LaTeX** is a de facto standard for research publications
- But not widely used in **school** or **undergraduate** math
- **Not** compact or easy to read or type (for non-technical users)

Aims of a convenient linear math notation

- Close to standard mathematics
Motto: if it **looks like math**, it should **work**
- Easy to read
- **Easy to type**
- Formulas should be **short**
- No obscure syntax errors
- Syntax easy to define and remember
- Mostly language independent
- Simple to extend or modify (localization)

Aims of a convenient linear math notation

- Close to standard mathematics

Motto: if it **looks like math**, it should **work**

- Easy to read

- **Easy to type**

- Formulas should be **short**

- No obscure syntax errors

- Syntax easy to define and remember

- Mostly language independent

- Simple to extend or modify (localization)

Aims of a convenient linear math notation

- Close to standard mathematics

Motto: if it **looks like math**, it should **work**

- Easy to read

- **Easy to type**

- Formulas should be **short**

- No obscure syntax errors

- Syntax easy to define and remember

- Mostly language independent

- Simple to extend or modify (localization)

Aims of a convenient linear math notation

- Close to standard mathematics
Motto: if it **looks like math**, it should **work**
- Easy to read
- **Easy to type**
- Formulas should be **short**
- No obscure syntax errors
- Syntax easy to define and remember
- Mostly language independent
- Simple to extend or modify (localization)

Aims of a convenient linear math notation

- Close to standard mathematics
Motto: if it **looks like math**, it should **work**
- Easy to read
- **Easy to type**
- Formulas should be **short**
- No obscure syntax errors
- Syntax easy to define and remember
- Mostly language independent
- Simple to extend or modify (localization)

Aims of a convenient linear math notation

- Close to standard mathematics
Motto: if it **looks like math**, it should **work**
- Easy to read
- **Easy to type**
- Formulas should be **short**
- No obscure syntax errors
- Syntax easy to define and remember
- Mostly language independent
- Simple to extend or modify (localization)

Aims of a convenient linear math notation

- Close to standard mathematics
Motto: if it **looks like math**, it should **work**
- Easy to read
- **Easy to type**
- Formulas should be **short**
- No obscure syntax errors
- Syntax easy to define and remember
- Mostly language independent
- Simple to extend or modify (localization)

Aims of a convenient linear math notation

- Close to standard mathematics
Motto: if it **looks like math**, it should **work**
- Easy to read
- **Easy to type**
- Formulas should be **short**
- No obscure syntax errors
- Syntax easy to define and remember
- Mostly language independent
- Simple to extend or modify (localization)

Aims of a convenient linear math notation

- Close to standard mathematics
Motto: if it **looks like math**, it should **work**
- Easy to read
- **Easy to type**
- Formulas should be **short**
- No obscure syntax errors
- Syntax easy to define and remember
- Mostly language independent
- Simple to extend or modify (localization)

Aims of a convenient linear math notation

- Close to standard mathematics
Motto: if it **looks like math**, it should **work**
- Easy to read
- **Easy to type**
- Formulas should be **short**
- No obscure syntax errors
- Syntax easy to define and remember
- Mostly language independent
- Simple to extend or modify (localization)

Aims of a convenient linear math notation

- Close to standard mathematics
Motto: if it **looks like math**, it should **work**
- Easy to read
- **Easy to type**
- Formulas should be **short**
- No obscure syntax errors
- Syntax easy to define and remember
- Mostly language independent
- Simple to extend or modify (localization)

ASCIIMath

A linear math notation with 8 syntax rules; designed in 2004

Based on well-known ASCII math conventions + some LaTeX

`c ::= [A-Za-z] | greek chr | numbers | ... constant symbols`

`u ::= sqrt | text | bb | ... prefix unary symbols`

`b ::= frac | root | stackrel prefix binary symbols`

`l ::= (| [| { | (: | {: left brackets`

`r ::=) |] | } | :) | :} right brackets`

`S ::= c | lEr | uS | bSS | "any" Simple expression`

`I ::= S_S | S^S | S_S^S | S Intermediate expression`

`E ::= IE | I/I Expression`

ASCIIMath

A linear math notation with 8 syntax rules; designed in 2004

Based on well-known ASCII math conventions + some LaTeX

`c ::= [A-Za-z] | greek chr | numbers | ... constant symbols`

`u ::= sqrt | text | bb | ... prefix unary symbols`

`b ::= frac | root | stackrel prefix binary symbols`

`l ::= (| [| { | (: | {: left brackets`

`r ::=) |] | } | :) | :} right brackets`

`S ::= c | lEr | uS | bSS | "any" Simple expression`

`I ::= S_S | S^S | S_S^S | S Intermediate expression`

`E ::= IE | I/I Expression`

ASCIIMath

A linear math notation with 8 syntax rules; designed in 2004

Based on well-known ASCII math conventions + some LaTeX

`c ::= [A-Za-z] | greek chr | numbers | ... constant symbols`

`u ::= sqrt | text | bb | ... prefix unary symbols`

`b ::= frac | root | stackrel prefix binary symbols`

`l ::= (| [| { | (: | {: left brackets`

`r ::=) |] | } | :) | :} right brackets`

`S ::= c | lEr | uS | bSS | "any" Simple expression`

`I ::= S_S | S^S | S_S^S | S Intermediate expression`

`E ::= IE | I/I Expression`

ASCIIMath

A linear math notation with 8 syntax rules; designed in 2004

Based on well-known ASCII math conventions + some LaTeX

`c ::= [A-Za-z] | greek chr | numbers | ... constant symbols`

`u ::= sqrt | text | bb | ... prefix unary symbols`

`b ::= frac | root | stackrel prefix binary symbols`

`l ::= (| [| { | (: | {: left brackets`

`r ::=) |] | } | :) | :} right brackets`

`S ::= c | lEr | uS | bSS | "any" Simple expression`

`I ::= S_S | S^S | S_S^S | S Intermediate expression`

`E ::= IE | I/I Expression`

ASCIIMath

A linear math notation with 8 syntax rules; designed in 2004

Based on well-known ASCII math conventions + some LaTeX

`c ::= [A-Za-z] | greek chr | numbers | ... constant symbols`

`u ::= sqrt | text | bb | ... prefix unary symbols`

`b ::= frac | root | stackrel prefix binary symbols`

`l ::= (| [| { | (: | {: left brackets`

`r ::=) |] | } | :) | :} right brackets`

`S ::= c | lEr | uS | bSS | "any" Simple expression`

`I ::= S_S | S^S | S_S^S | S Intermediate expression`

`E ::= IE | I/I Expression`

ASCIIMath

A linear math notation with 8 syntax rules; designed in 2004

Based on well-known ASCII math conventions + some LaTeX

`c ::= [A-Za-z] | greek chr | numbers | ... constant symbols`

`u ::= sqrt | text | bb | ... prefix unary symbols`

`b ::= frac | root | stackrel prefix binary symbols`

`l ::= (| [| { | (: | {: left brackets`

`r ::=) |] | } | :) | :} right brackets`

`S ::= c | lEr | uS | bSS | "any" Simple expression`

`I ::= S_S | S^S | S_S^S | S Intermediate expression`

`E ::= IE | I/I Expression`

ASCIIMath

A linear math notation with 8 syntax rules; designed in 2004

Based on well-known ASCII math conventions + some LaTeX

`c ::= [A-Za-z] | greek chr | numbers | ... constant symbols`

`u ::= sqrt | text | bb | ... prefix unary symbols`

`b ::= frac | root | stackrel prefix binary symbols`

`l ::= (| [| { | (: | {: left brackets`

`r ::=) |] | } | :) | :} right brackets`

`S ::= c | lEr | uS | bSS | "any" Simple expression`

`I ::= S_S | S^S | S_S^S | S Intermediate expression`

`E ::= IE | I/I Expression`

ASCIIMath

A linear math notation with 8 syntax rules; designed in 2004

Based on well-known ASCII math conventions + some LaTeX

`c ::= [A-Za-z] | greek chr | numbers | ... constant symbols`

`u ::= sqrt | text | bb | ... prefix unary symbols`

`b ::= frac | root | stackrel prefix binary symbols`

`l ::= (| [| { | (: | {: left brackets`

`r ::=) |] | } | :) | :} right brackets`

`S ::= c | lEr | uS | bSS | "any" Simple expression`

`I ::= S_S | S^S | S_S^S | S Intermediate expression`

`E ::= IE | I/I Expression`

ASCIIMath

A linear math notation with 8 syntax rules; designed in 2004

Based on well-known ASCII math conventions + some LaTeX

<code>c ::= [A-Za-z] greek chr numbers ...</code>	constant symbols
<code>u ::= sqrt text bb ...</code>	prefix unary symbols
<code>b ::= frac root stackrel</code>	prefix binary symbols
<code>l ::= ([{ (: {:</code>	left brackets
<code>r ::=)] } :) :}</code>	right brackets
<code>S ::= c lEr uS bSS "any"</code>	Simple expression
<code>I ::= S_S S^S S_S^S S</code>	Intermediate expression
<code>E ::= IE I/I</code>	Expression

ASCIIMath

A linear math notation with 8 syntax rules; designed in 2004

Based on well-known ASCII math conventions + some LaTeX

<code>c ::= [A-Za-z] greek chr numbers ...</code>	constant symbols
<code>u ::= sqrt text bb ...</code>	prefix unary symbols
<code>b ::= frac root stackrel</code>	prefix binary symbols
<code>l ::= ([{ (: {:</code>	left brackets
<code>r ::=)] } :) :}</code>	right brackets
<code>S ::= c lEr uS bSS "any"</code>	Simple expression
<code>I ::= S_S S^S S_S^S S</code>	Intermediate expression
<code>E ::= IE I/I</code>	Expression

Translation to MathML

- Each terminal symbol is translated to a corresponding MathML node
- Constants are mostly converted to their respective Unicode symbols
- $|Sr$ \rightarrow $\langle\text{mrow}\rangle|Sr\langle/\text{mrow}\rangle$ (brackets don't have to match)
- \sqrt{S} \rightarrow $\langle\text{msqrt}\rangle S' \langle/\text{msqrt}\rangle$
- "any" \rightarrow $\langle\text{mtext}\rangle\text{any}\langle/\text{mtext}\rangle$
- $\frac{S_1}{S_2}$ \rightarrow $\langle\text{mfrac}\rangle S'_1 S'_2 \langle/\text{mfrac}\rangle$
- $\sqrt[S_1]{S_2}$ \rightarrow $\langle\text{mroot}\rangle S'_2 S'_1 \langle/\text{mroot}\rangle$
- $S_1 S_2$ \rightarrow $\langle\text{mover}\rangle S'_2 S'_1 \langle/\text{mover}\rangle$
- S_1/S_2 \rightarrow $\langle\text{mfrac}\rangle S'_1 S'_2 \langle/\text{mfrac}\rangle$
- $S_1 S_2$ \rightarrow $\langle\text{msub}\rangle S_1 S'_2 \langle/\text{msub}\rangle$
- $S_1^S_2$ \rightarrow $\langle\text{msup}\rangle S_1 S'_2 \langle/\text{msup}\rangle$
- $S_1 S_2^S_3$ \rightarrow $\langle\text{msubsup}\rangle S_1 S'_2 S'_3 \langle/\text{msubsup}\rangle$ or $\langle\text{munderover}\rangle$
- Note: S' is the same as S , except that if S has an outer level of brackets, then S' is the expression inside these brackets

Translation to MathML

- Each terminal symbol is translated to a corresponding MathML node
- Constants are mostly converted to their respective Unicode symbols
- $|S_r| \rightarrow \langle \text{mrow} \rangle |S_r \langle / \text{mrow} \rangle$ (brackets don't have to match)
- $\sqrt{S} \rightarrow \langle \text{msqrt} \rangle S' \langle / \text{msqrt} \rangle$
- "any" $\rightarrow \langle \text{mtext} \rangle \text{any} \langle / \text{mtext} \rangle$
- $\frac{S_1}{S_2} \rightarrow \langle \text{mfrac} \rangle S'_1 S'_2 \langle / \text{mfrac} \rangle$
- $\sqrt[S_1]{S_2} \rightarrow \langle \text{mroot} \rangle S'_2 S'_1 \langle / \text{mroot} \rangle$
- $S_1 S_2 \rightarrow \langle \text{mover} \rangle S'_2 S'_1 \langle / \text{mover} \rangle$
- $S_1/S_2 \rightarrow \langle \text{mfrac} \rangle S'_1 S'_2 \langle / \text{mfrac} \rangle$
- $S_1 S_2 \rightarrow \langle \text{msub} \rangle S_1 S'_2 \langle / \text{msub} \rangle$
- $S_1^S_2 \rightarrow \langle \text{msup} \rangle S_1 S'_2 \langle / \text{msup} \rangle$
- $S_1 S_2^S_3 \rightarrow \langle \text{msubsup} \rangle S_1 S'_2 S'_3 \langle / \text{msubsup} \rangle$ or $\langle \text{munderover} \rangle$
- Note: S' is the same as S , except that if S has an outer level of brackets, then S' is the expression inside these brackets

Translation to MathML

- Each terminal symbol is translated to a corresponding MathML node
- Constants are mostly converted to their respective Unicode symbols
- $|Sr \rightarrow \langle mrow \rangle |Sr \langle /mrow \rangle$ (brackets don't have to match)
- $\sqrt{S} \rightarrow \langle msqrt \rangle S' \langle /msqrt \rangle$
- "any" $\rightarrow \langle mtext \rangle any \langle /mtext \rangle$
- $\frac{S_1}{S_2} \rightarrow \langle mfrac \rangle S'_1 S'_2 \langle /mfrac \rangle$
- $\sqrt[S_1]{S_2} \rightarrow \langle mroot \rangle S'_2 S'_1 \langle /mroot \rangle$
- $S_1 S_2 \rightarrow \langle mover \rangle S'_2 S'_1 \langle /mover \rangle$
- $S_1/S_2 \rightarrow \langle mfrac \rangle S'_1 S'_2 \langle /mfrac \rangle$
- $S_1 S_2 \rightarrow \langle msub \rangle S_1 S'_2 \langle /msub \rangle$
- $S_1^S_2 \rightarrow \langle msup \rangle S_1 S'_2 \langle /msup \rangle$
- $S_1 S_2^S_3 \rightarrow \langle msubsup \rangle S_1 S'_2 S'_3 \langle /msubsup \rangle$ or $\langle munderover \rangle$
- Note: S' is the same as S , except that if S has an outer level of brackets, then S' is the expression inside these brackets

Translation to MathML

- Each terminal symbol is translated to a corresponding MathML node
- Constants are mostly converted to their respective Unicode symbols
- $ISr \rightarrow \langle mrow \rangle ISr \langle /mrow \rangle$ (brackets don't have to match)
- $\sqrt{S} \rightarrow \langle msqrt \rangle S' \langle /msqrt \rangle$
- "any" $\rightarrow \langle mtext \rangle any \langle /mtext \rangle$
- $\frac{S_1}{S_2} \rightarrow \langle mfrac \rangle S'_1 S'_2 \langle /mfrac \rangle$
- $\sqrt[S_1]{S_2} \rightarrow \langle mroot \rangle S'_2 S'_1 \langle /mroot \rangle$
- $\frac{S_1}{S_2} S_3 \rightarrow \langle mover \rangle S'_2 S'_1 \langle /mover \rangle$
- $\frac{S_1}{S_2} \rightarrow \langle mfrac \rangle S'_1 S'_2 \langle /mfrac \rangle$
- $S_1 S_2 \rightarrow \langle msub \rangle S_1 S'_2 \langle /msub \rangle$
- $S_1^S \rightarrow \langle msup \rangle S_1 S'_2 \langle /msup \rangle$
- $S_1 S_2^S S_3 \rightarrow \langle msubsup \rangle S_1 S'_2 S'_3 \langle /msubsup \rangle$ or $\langle munderover \rangle$
- Note: S' is the same as S , except that if S has an outer level of brackets, then S' is the expression inside these brackets

Translation to MathML

- Each terminal symbol is translated to a corresponding MathML node
- Constants are mostly converted to their respective Unicode symbols
- $|Sr \rightarrow \langle \text{mrow} \rangle |Sr \langle / \text{mrow} \rangle$ (brackets don't have to match)
- $\sqrt{S} \rightarrow \langle \text{msqrt} \rangle S' \langle / \text{msqrt} \rangle$
- "any" $\rightarrow \langle \text{mtext} \rangle \text{any} \langle / \text{mtext} \rangle$
- $\frac{S_1}{S_2} \rightarrow \langle \text{mfrac} \rangle S'_1 S'_2 \langle / \text{mfrac} \rangle$
- $\sqrt[S_1]{S_2} \rightarrow \langle \text{mroot} \rangle S'_2 S'_1 \langle / \text{mroot} \rangle$
- $S_1 S_2 \rightarrow \langle \text{mover} \rangle S'_2 S'_1 \langle / \text{mover} \rangle$
- $S_1/S_2 \rightarrow \langle \text{mfrac} \rangle S'_1 S'_2 \langle / \text{mfrac} \rangle$
- $S_1 S_2 \rightarrow \langle \text{msub} \rangle S_1 S'_2 \langle / \text{msub} \rangle$
- $S_1^S \rightarrow \langle \text{msup} \rangle S_1 S'_2 \langle / \text{msup} \rangle$
- $S_1 S_2^S \rightarrow \langle \text{msubsup} \rangle S_1 S'_2 S'_3 \langle / \text{msubsup} \rangle$ or $\langle \text{munderover} \rangle$
- Note: S' is the same as S , except that if S has an outer level of brackets, then S' is the expression inside these brackets

Translation to MathML

- Each terminal symbol is translated to a corresponding MathML node
- Constants are mostly converted to their respective Unicode symbols
- $|Sr \rightarrow \langle \text{mrow} \rangle |Sr \langle / \text{mrow} \rangle$ (brackets don't have to match)
- $\sqrt{S} \rightarrow \langle \text{msqrt} \rangle S' \langle / \text{msqrt} \rangle$
- "any" $\rightarrow \langle \text{mtext} \rangle \text{any} \langle / \text{mtext} \rangle$
- $\frac{S_1}{S_2} \rightarrow \langle \text{mfrac} \rangle S'_1 S'_2 \langle / \text{mfrac} \rangle$
- $\sqrt[S_1]{S_2} \rightarrow \langle \text{mroot} \rangle S'_2 S'_1 \langle / \text{mroot} \rangle$
- $S_1 S_2 \rightarrow \langle \text{mover} \rangle S'_2 S'_1 \langle / \text{mover} \rangle$
- $S_1/S_2 \rightarrow \langle \text{mfrac} \rangle S'_1 S'_2 \langle / \text{mfrac} \rangle$
- $S_1 S_2 \rightarrow \langle \text{msub} \rangle S_1 S'_2 \langle / \text{msub} \rangle$
- $S_1^S_2 \rightarrow \langle \text{msup} \rangle S_1 S'_2 \langle / \text{msup} \rangle$
- $S_1 S_2^S_3 \rightarrow \langle \text{msubsup} \rangle S_1 S'_2 S'_3 \langle / \text{msubsup} \rangle$ or $\langle \text{munderover} \rangle$
- Note: S' is the same as S , except that if S has an outer level of brackets, then S' is the expression inside these brackets

Translation to MathML

- Each terminal symbol is translated to a corresponding MathML node
- Constants are mostly converted to their respective Unicode symbols
- $|Sr$ \rightarrow $\langle\text{mrow}\rangle|Sr\langle/\text{mrow}\rangle$ (brackets don't have to match)
- \sqrt{S} \rightarrow $\langle\text{msqrt}\rangle S' \langle/\text{msqrt}\rangle$
- "any" \rightarrow $\langle\text{mtext}\rangle\text{any}\langle/\text{mtext}\rangle$
- $\frac{S_1}{S_2}$ \rightarrow $\langle\text{mfrac}\rangle S'_1 S'_2 \langle/\text{mfrac}\rangle$
- $\sqrt[S_1]{S_2}$ \rightarrow $\langle\text{mroot}\rangle S'_2 S'_1 \langle/\text{mroot}\rangle$
- $S_1 S_2$ \rightarrow $\langle\text{mover}\rangle S'_2 S'_1 \langle/\text{mover}\rangle$
- S_1/S_2 \rightarrow $\langle\text{mfrac}\rangle S'_1 S'_2 \langle/\text{mfrac}\rangle$
- $S_1 S_2$ \rightarrow $\langle\text{msub}\rangle S_1 S'_2 \langle/\text{msub}\rangle$
- S_1^S \rightarrow $\langle\text{msup}\rangle S_1 S'_2 \langle/\text{msup}\rangle$
- $S_1 S_2^S S_3$ \rightarrow $\langle\text{msubsup}\rangle S_1 S'_2 S'_3 \langle/\text{msubsup}\rangle$ or $\langle\text{munderover}\rangle$
- Note: S' is the same as S , except that if S has an outer level of brackets, then S' is the expression inside these brackets

Translation to MathML

- Each terminal symbol is translated to a corresponding MathML node
- Constants are mostly converted to their respective Unicode symbols
- $|Sr \rightarrow \langle \text{mrow} \rangle |Sr \langle / \text{mrow} \rangle$ (brackets don't have to match)
- $\sqrt{S} \rightarrow \langle \text{msqrt} \rangle S' \langle / \text{msqrt} \rangle$
- "any" $\rightarrow \langle \text{mtext} \rangle \text{any} \langle / \text{mtext} \rangle$
- $\frac{S_1}{S_2} \rightarrow \langle \text{mfrac} \rangle S'_1 S'_2 \langle / \text{mfrac} \rangle$
- $\sqrt[S_1]{S_2} \rightarrow \langle \text{mroot} \rangle S'_2 S'_1 \langle / \text{mroot} \rangle$
- $S_1 S_2 \rightarrow \langle \text{mover} \rangle S'_2 S'_1 \langle / \text{mover} \rangle$
- $S_1/S_2 \rightarrow \langle \text{mfrac} \rangle S'_1 S'_2 \langle / \text{mfrac} \rangle$
- $S_1 S_2 \rightarrow \langle \text{msub} \rangle S_1 S'_2 \langle / \text{msub} \rangle$
- $S_1^S_2 \rightarrow \langle \text{msup} \rangle S_1 S'_2 \langle / \text{msup} \rangle$
- $S_1 S_2^S_3 \rightarrow \langle \text{msubsup} \rangle S_1 S'_2 S'_3 \langle / \text{msubsup} \rangle$ or $\langle \text{munderover} \rangle$
- Note: S' is the same as S , except that if S has an outer level of brackets, then S' is the expression inside these brackets

Translation to MathML

- Each terminal symbol is translated to a corresponding MathML node
- Constants are mostly converted to their respective Unicode symbols
- $|Sr$ \rightarrow $\langle \text{mrow} \rangle |Sr \langle / \text{mrow} \rangle$ (brackets don't have to match)
- \sqrt{S} \rightarrow $\langle \text{msqrt} \rangle S' \langle / \text{msqrt} \rangle$
- "any" \rightarrow $\langle \text{mtext} \rangle \text{any} \langle / \text{mtext} \rangle$
- $\frac{S_1}{S_2}$ \rightarrow $\langle \text{mfrac} \rangle S'_1 S'_2 \langle / \text{mfrac} \rangle$
- $\sqrt[S_1]{S_2}$ \rightarrow $\langle \text{mroot} \rangle S'_2 S'_1 \langle / \text{mroot} \rangle$
- $S_1 S_2$ \rightarrow $\langle \text{mover} \rangle S'_2 S'_1 \langle / \text{mover} \rangle$
- S_1/S_2 \rightarrow $\langle \text{mfrac} \rangle S'_1 S'_2 \langle / \text{mfrac} \rangle$
- $S_1 S_2$ \rightarrow $\langle \text{msub} \rangle S_1 S'_2 \langle / \text{msub} \rangle$
- S_1^S \rightarrow $\langle \text{msup} \rangle S_1 S'_2 \langle / \text{msup} \rangle$
- $S_1 S_2^S S_3$ \rightarrow $\langle \text{msubsup} \rangle S_1 S'_2 S'_3 \langle / \text{msubsup} \rangle$ or $\langle \text{munderover} \rangle$
- Note: S' is the same as S , except that if S has an outer level of brackets, then S' is the expression inside these brackets

Translation to MathML

- Each terminal symbol is translated to a corresponding MathML node
- Constants are mostly converted to their respective Unicode symbols
- $|Sr$ \rightarrow $\langle \text{mrow} \rangle |Sr \langle / \text{mrow} \rangle$ (brackets don't have to match)
- \sqrt{S} \rightarrow $\langle \text{msqrt} \rangle S' \langle / \text{msqrt} \rangle$
- "any" \rightarrow $\langle \text{mtext} \rangle \text{any} \langle / \text{mtext} \rangle$
- $\frac{S_1}{S_2}$ \rightarrow $\langle \text{mfrac} \rangle S'_1 S'_2 \langle / \text{mfrac} \rangle$
- $\sqrt[S_1]{S_2}$ \rightarrow $\langle \text{mroot} \rangle S'_2 S'_1 \langle / \text{mroot} \rangle$
- $S_1 S_2$ \rightarrow $\langle \text{mover} \rangle S'_2 S'_1 \langle / \text{mover} \rangle$
- S_1/S_2 \rightarrow $\langle \text{mfrac} \rangle S'_1 S'_2 \langle / \text{mfrac} \rangle$
- $S_1 S_2$ \rightarrow $\langle \text{msub} \rangle S_1 S'_2 \langle / \text{msub} \rangle$
- $S_1^S_2$ \rightarrow $\langle \text{msup} \rangle S_1 S'_2 \langle / \text{msup} \rangle$
- $S_1 S_2^S_3$ \rightarrow $\langle \text{msubsup} \rangle S_1 S'_2 S'_3 \langle / \text{msubsup} \rangle$ or $\langle \text{munderover} \rangle$
- Note: S' is the same as S , except that if S has an outer level of brackets, then S' is the expression inside these brackets

Translation to MathML

- Each terminal symbol is translated to a corresponding MathML node
- Constants are mostly converted to their respective Unicode symbols
- $|Sr \rightarrow \langle \text{mrow} \rangle |Sr \langle / \text{mrow} \rangle$ (brackets don't have to match)
- $\sqrt{S} \rightarrow \langle \text{msqrt} \rangle S' \langle / \text{msqrt} \rangle$
- "any" $\rightarrow \langle \text{mtext} \rangle \text{any} \langle / \text{mtext} \rangle$
- $\frac{S_1}{S_2} \rightarrow \langle \text{mfrac} \rangle S'_1 S'_2 \langle / \text{mfrac} \rangle$
- $\sqrt[S_1]{S_2} \rightarrow \langle \text{mroot} \rangle S'_2 S'_1 \langle / \text{mroot} \rangle$
- $S_1 S_2 \rightarrow \langle \text{mover} \rangle S'_2 S'_1 \langle / \text{mover} \rangle$
- $S_1/S_2 \rightarrow \langle \text{mfrac} \rangle S'_1 S'_2 \langle / \text{mfrac} \rangle$
- $S_1 S_2 \rightarrow \langle \text{msub} \rangle S_1 S'_2 \langle / \text{msub} \rangle$
- $S_1^{\wedge} S_2 \rightarrow \langle \text{msup} \rangle S_1 S'_2 \langle / \text{msup} \rangle$
- $S_1 S_2^{\wedge} S_3 \rightarrow \langle \text{msubsup} \rangle S_1 S'_2 S'_3 \langle / \text{msubsup} \rangle$ or $\langle \text{munderover} \rangle$
- Note: S' is the same as S , except that if S has an outer level of brackets, then S' is the expression inside these brackets

Translation to MathML

- Each terminal symbol is translated to a corresponding MathML node
- Constants are mostly converted to their respective Unicode symbols
- $|Sr \rightarrow \langle \text{mrow} \rangle |Sr \langle / \text{mrow} \rangle$ (brackets don't have to match)
- $\sqrt{S} \rightarrow \langle \text{msqrt} \rangle S' \langle / \text{msqrt} \rangle$
- "any" $\rightarrow \langle \text{mtext} \rangle \text{any} \langle / \text{mtext} \rangle$
- $\frac{S_1}{S_2} \rightarrow \langle \text{mfrac} \rangle S'_1 S'_2 \langle / \text{mfrac} \rangle$
- $\sqrt[S_1]{S_2} \rightarrow \langle \text{mroot} \rangle S'_2 S'_1 \langle / \text{mroot} \rangle$
- $S_1 S_2 \rightarrow \langle \text{mover} \rangle S'_2 S'_1 \langle / \text{mover} \rangle$
- $S_1/S_2 \rightarrow \langle \text{mfrac} \rangle S'_1 S'_2 \langle / \text{mfrac} \rangle$
- $S_1 S_2 \rightarrow \langle \text{msub} \rangle S_1 S'_2 \langle / \text{msub} \rangle$
- $S_1^S \rightarrow \langle \text{msup} \rangle S_1 S'_2 \langle / \text{msup} \rangle$
- $S_1 S_2^S S_3 \rightarrow \langle \text{msubsup} \rangle S_1 S'_2 S'_3 \langle / \text{msubsup} \rangle$ or $\langle \text{munderover} \rangle$
- Note: S' is the same as S , except that if S has an outer level of brackets, then S' is the expression inside these brackets

Translation to MathML

- Each terminal symbol is translated to a corresponding MathML node
- Constants are mostly converted to their respective Unicode symbols
- $|Sr \rightarrow \langle \text{mrow} \rangle |Sr \langle / \text{mrow} \rangle$ (brackets don't have to match)
- $\sqrt{S} \rightarrow \langle \text{msqrt} \rangle S' \langle / \text{msqrt} \rangle$
- "any" $\rightarrow \langle \text{mtext} \rangle \text{any} \langle / \text{mtext} \rangle$
- $\frac{S_1}{S_2} \rightarrow \langle \text{mfrac} \rangle S'_1 S'_2 \langle / \text{mfrac} \rangle$
- $\sqrt[S_1]{S_2} \rightarrow \langle \text{mroot} \rangle S'_2 S'_1 \langle / \text{mroot} \rangle$
- $S_1 S_2 \rightarrow \langle \text{mover} \rangle S'_2 S'_1 \langle / \text{mover} \rangle$
- $S_1/S_2 \rightarrow \langle \text{mfrac} \rangle S'_1 S'_2 \langle / \text{mfrac} \rangle$
- $S_1 S_2 \rightarrow \langle \text{msub} \rangle S_1 S'_2 \langle / \text{msub} \rangle$
- $S_1^{\wedge} S_2 \rightarrow \langle \text{msup} \rangle S_1 S'_2 \langle / \text{msup} \rangle$
- $S_1 S_2^{\wedge} S_3 \rightarrow \langle \text{msubsup} \rangle S_1 S'_2 S'_3 \langle / \text{msubsup} \rangle$ or $\langle \text{munderover} \rangle$
- Note: S' is the same as S , except that if S has an outer level of brackets, then S' is the expression inside these brackets

Translation to MathML

- Each terminal symbol is translated to a corresponding MathML node
- Constants are mostly converted to their respective Unicode symbols
- $|Sr \rightarrow \langle \text{mrow} \rangle |Sr \langle / \text{mrow} \rangle$ (brackets don't have to match)
- $\sqrt{S} \rightarrow \langle \text{msqrt} \rangle S' \langle / \text{msqrt} \rangle$
- "any" $\rightarrow \langle \text{mtext} \rangle \text{any} \langle / \text{mtext} \rangle$
- $\frac{S_1}{S_2} \rightarrow \langle \text{mfrac} \rangle S'_1 S'_2 \langle / \text{mfrac} \rangle$
- $\sqrt[S_1]{S_2} \rightarrow \langle \text{mroot} \rangle S'_2 S'_1 \langle / \text{mroot} \rangle$
- $S_1 S_2 \rightarrow \langle \text{mover} \rangle S'_2 S'_1 \langle / \text{mover} \rangle$
- $S_1/S_2 \rightarrow \langle \text{mfrac} \rangle S'_1 S'_2 \langle / \text{mfrac} \rangle$
- $S_1 S_2 \rightarrow \langle \text{msub} \rangle S_1 S'_2 \langle / \text{msub} \rangle$
- $S_1^S \rightarrow \langle \text{msup} \rangle S_1 S'_2 \langle / \text{msup} \rangle$
- $S_1 S_2^S \rightarrow \langle \text{msubsup} \rangle S_1 S'_2 S'_3 \langle / \text{msubsup} \rangle$ or $\langle \text{munderover} \rangle$
- Note: S' is the same as S , except that if S has an outer level of brackets, then S' is the expression inside these brackets

Examples of ASCIIMath

$$\lim_{x \rightarrow \infty} \tan^{-1} x = \frac{\pi}{2}$$

$$\lim_{(x \rightarrow \infty)} \tan^{-1} x = \text{pi}/2$$

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$$

$$\text{sum}_{(n=1)}^{\infty} 1/n^2 = \text{pi}^2/6$$

$$\int_{-1}^1 \sqrt{1-x^2} dx = \frac{\pi}{2}$$

$$\text{int}_{-1}^1 \text{sqrt}(1-x^2) dx = \text{pi}/2$$

$$[0, 1) = \{x \in \mathbb{R} : 0 \leq x < 1\}$$

$$[0, 1) = \{x \text{ in } \mathbb{R} : 0 \leq x < 1\}$$

These examples carry students a long way

Examples of ASCIIMath

$$\lim_{x \rightarrow \infty} \tan^{-1} x = \frac{\pi}{2}$$

$$\lim_{(x \rightarrow \infty)} \tan^{-1} x = \pi/2$$

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$$

$$\text{sum}_{(n=1)}^{\infty} 1/n^2 = \pi^2/6$$

$$\int_{-1}^1 \sqrt{1-x^2} dx = \frac{\pi}{2}$$

$$\text{int}_{-1}^1 \text{sqrt}(1-x^2) dx = \pi/2$$

$$[0, 1) = \{x \in \mathbb{R} : 0 \leq x < 1\}$$

$$[0, 1) = \{x \text{ in } \mathbb{R} : 0 \leq x < 1\}$$

These examples carry students a long way

Examples of ASCIIMath

$$\lim_{x \rightarrow \infty} \tan^{-1} x = \frac{\pi}{2}$$

$$\lim_{(x \rightarrow \infty)} \tan^{-1} x = \text{pi}/2$$

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$$

$$\text{sum}_{(n=1)}^{\infty} 1/n^2 = \text{pi}^2/6$$

$$\int_{-1}^1 \sqrt{1-x^2} dx = \frac{\pi}{2}$$

$$\text{int}_{-1}^1 \text{sqrt}(1-x^2) dx = \text{pi}/2$$

$$[0, 1) = \{x \in \mathbb{R} : 0 \leq x < 1\}$$

$$[0, 1) = \{x \text{ in } \mathbb{R} : 0 \leq x < 1\}$$

These examples carry students a long way

Examples of ASCIIMath

$$\lim_{x \rightarrow \infty} \tan^{-1} x = \frac{\pi}{2}$$

$$\lim_{(x \rightarrow \infty)} \tan^{-1} x = \text{pi}/2$$

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$$

$$\text{sum}_{(n=1)}^{\infty} 1/n^2 = \text{pi}^2/6$$

$$\int_{-1}^1 \sqrt{1-x^2} dx = \frac{\pi}{2}$$

$$\text{int}_{-1}^1 \text{sqrt}(1-x^2) dx = \text{pi}/2$$

$$[0, 1) = \{x \in \mathbb{R} : 0 \leq x < 1\}$$

$$[0, 1) = \{x \text{ in } \mathbb{R} : 0 \leq x < 1\}$$

These examples carry students a long way

Examples of ASCIIMath

$$\lim_{x \rightarrow \infty} \tan^{-1} x = \frac{\pi}{2}$$

$$\lim_{(x \rightarrow \infty)} \tan^{-1} x = \pi/2$$

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$$

$$\text{sum}_{(n=1)}^{\infty} 1/n^2 = \pi^2/6$$

$$\int_{-1}^1 \sqrt{1-x^2} dx = \frac{\pi}{2}$$

$$\text{int}_{-1}^1 \text{sqrt}(1-x^2) dx = \pi/2$$

$$[0, 1) = \{x \in \mathbb{R} : 0 \leq x < 1\}$$

$$[0, 1) = \{x \text{ in } \mathbb{R} : 0 \leq x < 1\}$$

These examples carry students a long way

Examples of ASCIIMath

$$\lim_{x \rightarrow \infty} \tan^{-1} x = \frac{\pi}{2}$$

$$\lim_{(x \rightarrow \infty)} \tan^{-1} x = \pi/2$$

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$$

$$\text{sum}_{(n=1)}^{\infty} 1/n^2 = \pi^2/6$$

$$\int_{-1}^1 \sqrt{1-x^2} dx = \frac{\pi}{2}$$

$$\text{int}_{-1}^1 \text{sqrt}(1-x^2) dx = \pi/2$$

$$[0, 1) = \{x \in \mathbb{R} : 0 \leq x < 1\}$$

$$[0, 1) = \{x \text{ in } \mathbb{R} : 0 \leq x < 1\}$$

These examples carry students a long way

Examples of ASCIIMath

$$\lim_{x \rightarrow \infty} \tan^{-1} x = \frac{\pi}{2}$$

$$\lim_{(x \rightarrow \infty)} \tan^{-1} x = \text{pi}/2$$

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$$

$$\text{sum}_{(n=1)}^{\infty} 1/n^2 = \text{pi}^2/6$$

$$\int_{-1}^1 \sqrt{1-x^2} dx = \frac{\pi}{2}$$

$$\text{int}_{-1}^1 \text{sqrt}(1-x^2) dx = \text{pi}/2$$

$$[0, 1) = \{x \in \mathbb{R} : 0 \leq x < 1\}$$

$$[0, 1) = \{x \text{ in } \mathbb{R} : 0 \leq x < 1\}$$

These examples carry students a long way

Examples of ASCIIMath

$$\lim_{x \rightarrow \infty} \tan^{-1} x = \frac{\pi}{2}$$

$$\lim_{(x \rightarrow \infty)} \tan^{-1} x = \text{pi}/2$$

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$$

$$\text{sum}_{(n=1)}^{\infty} 1/n^2 = \text{pi}^2/6$$

$$\int_{-1}^1 \sqrt{1-x^2} dx = \frac{\pi}{2}$$

$$\text{int}_{-1}^1 \text{sqrt}(1-x^2) dx = \text{pi}/2$$

$$[0, 1) = \{x \in \mathbb{R} : 0 \leq x < 1\}$$

$$[0, 1) = \{x \text{ in } \mathbb{R} : 0 \leq x < 1\}$$

These examples carry students a long way

Other features of ASCIIMath

- Tokenized by “longest initial matching substring”
- Non-matching letters are parsed as individual variables
- Grouping brackets do not have to match: x in $(a,b]$
- Math is delimited by ‘...’ (or more distracting $\$...\$$)
- Brackets removed if displayed formula can be parsed without
- No syntax errors (lowers the learning curve)
- Tokens are chosen to mimic how symbols are written by hand
- e.g. $\sim\sim$ for \approx , $O/$ for \emptyset , $+ -$ for \pm , RR for \mathbb{R}
- ASCIIMath also overlaps substantially with LaTeX
- relatively easy to switch from one language to the other
- Simple syntax for matrices: $[[1,2],[3,4]]$ or $((a,b),(c,d))$

Other features of ASCIIMath

- Tokenized by “longest initial matching substring”
- Non-matching letters are parsed as individual variables
- Grouping brackets do not have to match: x in $(a,b]$
- Math is delimited by ‘...’ (or more distracting $\$...\$$)
- Brackets removed if displayed formula can be parsed without
- No syntax errors (lowers the learning curve)
- Tokens are chosen to mimic how symbols are written by hand
- e.g. $\sim\sim$ for \approx , $O/$ for \emptyset , $+ -$ for \pm , RR for \mathbb{R}
- ASCIIMath also overlaps substantially with LaTeX
- relatively easy to switch from one language to the other
- Simple syntax for matrices: $[[1,2],[3,4]]$ or $((a,b),(c,d))$

Other features of ASCIIMath

- Tokenized by “longest initial matching substring”
- Non-matching letters are parsed as individual variables
- Grouping brackets do not have to match: x in (a,b)
- Math is delimited by ‘...’ (or more distracting $\$...\$$)
- Brackets removed if displayed formula can be parsed without
- No syntax errors (lowers the learning curve)
- Tokens are chosen to mimic how symbols are written by hand
- e.g. $\sim\sim$ for \approx , $O/$ for \emptyset , $+ -$ for \pm , RR for \mathbb{R}
- ASCIIMath also overlaps substantially with LaTeX
- relatively easy to switch from one language to the other
- Simple syntax for matrices: $[[1,2],[3,4]]$ or $((a,b),(c,d))$

Other features of ASCIIMath

- Tokenized by “longest initial matching substring”
- Non-matching letters are parsed as individual variables
- Grouping brackets do not have to match: x in (a,b)
- Math is delimited by ‘...’ (or more distracting $\$...\$$)
- Brackets removed if displayed formula can be parsed without
- No syntax errors (lowers the learning curve)
- Tokens are chosen to mimic how symbols are written by hand
- e.g. $\sim\sim$ for \approx , $O/$ for \emptyset , $+ -$ for \pm , RR for \mathbb{R}
- ASCIIMath also overlaps substantially with LaTeX
- relatively easy to switch from one language to the other
- Simple syntax for matrices: $[[1,2],[3,4]]$ or $((a,b),(c,d))$

Other features of ASCIIMath

- Tokenized by “longest initial matching substring”
- Non-matching letters are parsed as individual variables
- Grouping brackets do not have to match: x in (a,b)
- Math is delimited by ‘...’ (or more distracting $\$...\$$)
- Brackets removed if displayed formula can be parsed without
- No syntax errors (lowers the learning curve)
- Tokens are chosen to mimic how symbols are written by hand
- e.g. $\sim\sim$ for \approx , $O/$ for \emptyset , $+ -$ for \pm , RR for \mathbb{R}
- ASCIIMath also overlaps substantially with LaTeX
- relatively easy to switch from one language to the other
- Simple syntax for matrices: $[[1,2],[3,4]]$ or $((a,b),(c,d))$

Other features of ASCIIMath

- Tokenized by “longest initial matching substring”
- Non-matching letters are parsed as individual variables
- Grouping brackets do not have to match: x in (a,b)
- Math is delimited by ‘...’ (or more distracting $\$...\$$)
- Brackets removed if displayed formula can be parsed without
 - No syntax errors (lowers the learning curve)
 - Tokens are chosen to mimic how symbols are written by hand
 - e.g. $\sim\sim$ for \approx , $O/$ for \emptyset , $+ -$ for \pm , RR for \mathbb{R}
 - ASCIIMath also overlaps substantially with LaTeX
 - relatively easy to switch from one language to the other
 - Simple syntax for matrices: $[[1,2],[3,4]]$ or $((a,b),(c,d))$

Other features of ASCIIMath

- Tokenized by “longest initial matching substring”
- Non-matching letters are parsed as individual variables
- Grouping brackets do not have to match: x in (a,b)
- Math is delimited by ‘...’ (or more distracting $\$...\$$)
- Brackets removed if displayed formula can be parsed without
- No syntax errors (lowers the learning curve)
- Tokens are chosen to mimic how symbols are written by hand
- e.g. $\sim\sim$ for \approx , $O/$ for \emptyset , $+ -$ for \pm , RR for \mathbb{R}
- ASCIIMath also overlaps substantially with LaTeX
- relatively easy to switch from one language to the other
- Simple syntax for matrices: $[[1,2],[3,4]]$ or $((a,b),(c,d))$

Other features of ASCIIMath

- Tokenized by “longest initial matching substring”
- Non-matching letters are parsed as individual variables
- Grouping brackets do not have to match: x in (a,b)
- Math is delimited by ‘...’ (or more distracting $\$...\$$)
- Brackets removed if displayed formula can be parsed without
- No syntax errors (lowers the learning curve)
- Tokens are chosen to mimic how symbols are written by hand
- e.g. $\sim\sim$ for \approx , $O/$ for \emptyset , $+ -$ for \pm , RR for \mathbb{R}
- ASCIIMath also overlaps substantially with LaTeX
- relatively easy to switch from one language to the other
- Simple syntax for matrices: $[[1,2],[3,4]]$ or $((a,b),(c,d))$

Other features of ASCIIMath

- Tokenized by “longest initial matching substring”
- Non-matching letters are parsed as individual variables
- Grouping brackets do not have to match: x in (a,b)
- Math is delimited by ‘...’ (or more distracting $\$...\$$)
- Brackets removed if displayed formula can be parsed without
- No syntax errors (lowers the learning curve)
- Tokens are chosen to mimic how symbols are written by hand
- e.g. $\sim\sim$ for \approx , $O/$ for \emptyset , $+ -$ for \pm , RR for \mathbb{R}
- ASCIIMath also overlaps substantially with LaTeX
- relatively easy to switch from one language to the other
- Simple syntax for matrices: $[[1,2],[3,4]]$ or $((a,b),(c,d))$

Other features of ASCIIMath

- Tokenized by “longest initial matching substring”
- Non-matching letters are parsed as individual variables
- Grouping brackets do not have to match: x in (a,b)
- Math is delimited by ‘...’ (or more distracting $\$...\$$)
- Brackets removed if displayed formula can be parsed without
- No syntax errors (lowers the learning curve)
- Tokens are chosen to mimic how symbols are written by hand
- e.g. $\sim\sim$ for \approx , $O/$ for \emptyset , $+ -$ for \pm , RR for \mathbb{R}
- ASCIIMath also overlaps substantially with LaTeX
 - relatively easy to switch from one language to the other
 - Simple syntax for matrices: $[[1,2],[3,4]]$ or $((a,b),(c,d))$

Other features of ASCIIMath

- Tokenized by “longest initial matching substring”
- Non-matching letters are parsed as individual variables
- Grouping brackets do not have to match: x in (a,b)
- Math is delimited by ‘...’ (or more distracting $\$...\$$)
- Brackets removed if displayed formula can be parsed without
- No syntax errors (lowers the learning curve)
- Tokens are chosen to mimic how symbols are written by hand
- e.g. $\sim\sim$ for \approx , $O/$ for \emptyset , $+ -$ for \pm , RR for \mathbb{R}
- ASCIIMath also overlaps substantially with LaTeX
- relatively easy to switch from one language to the other
- Simple syntax for matrices: $[[1,2],[3,4]]$ or $((a,b),(c,d))$

Other features of ASCIIMath

- Tokenized by “longest initial matching substring”
- Non-matching letters are parsed as individual variables
- Grouping brackets do not have to match: x in (a,b)
- Math is delimited by ‘...’ (or more distracting $\$...\$$)
- Brackets removed if displayed formula can be parsed without
- No syntax errors (lowers the learning curve)
- Tokens are chosen to mimic how symbols are written by hand
- e.g. $\sim\sim$ for \approx , $O/$ for \emptyset , $+ -$ for \pm , RR for \mathbb{R}
- ASCIIMath also overlaps substantially with LaTeX
- relatively easy to switch from one language to the other
- Simple syntax for matrices: $[[1,2],[3,4]]$ or $((a,b),(c,d))$

How to use ASCIIMathML.js

- Implemented in a single JavaScript file, < 900 lines
- Conversion to MathML is done as the web page loads
- Makes MathML work in **HTML** in Firefox **and** IE

```
<html>
<head>
<script type="text/javascript" src="ASCIIMathML.js"></script>
</head>
<body>
Some formulas: 'sum_(i=1)^n i=(n(n+1))/2'
and  $\int_0^{\frac{\pi}{2}} \sin x \, dx=1$ .
</body>
</html>
```

How to use ASCIIMathML.js

- Implemented in a single JavaScript file, < 900 lines
- Conversion to MathML is done as the web page loads
- Makes MathML work in **HTML** in Firefox **and** IE

```
<html>
<head>
<script type="text/javascript" src="ASCIIMathML.js"></script>
</head>
<body>
Some formulas: 'sum_(i=1)^n i=(n(n+1))/2'
and  $\int_0^{\frac{\pi}{2}} \sin x \, dx = 1$ .
</body>
</html>
```

ASCIIMath is widely used

- Downloaded by thousands of users around the globe
- Integrated into many wikis, blogs, course management systems
- Augmented with ASCIIsvg and a JS scientific calculator [J 2004]
- Merged into WYSIWYG web editors HTMLArea, Xinha [J, Lippman 2006]
- Added to TiddlyWiki (client-side wiki) → ASciencePad
- ASCIIMath serverside in PHP [Chan 2004], Perl [Nodine 2006]
- Modified to LaTeXMathML.js [Woodall 2006]

Demo of ASciencePad

ASCIIMath is widely used

- Downloaded by thousands of users around the globe
- Integrated into many wikis, blogs, course management systems
- Augmented with ASCIISvg and a JS scientific calculator [J 2004]
- Merged into WYSIWYG web editors HTMLArea, Xinha [J, Lippman 2006]
- Added to TiddlyWiki (client-side wiki) → ASciencePad
- ASCIIMath serverside in PHP [Chan 2004], Perl [Nodine 2006]
- Modified to LaTeXMathML.js [Woodall 2006]

Demo of ASciencePad

ASCIIMath is widely used

- Downloaded by thousands of users around the globe
- Integrated into many wikis, blogs, course management systems
- Augmented with ASCIISvg and a JS scientific calculator [J 2004]
- Merged into WYSIWYG web editors HTMLArea, Xinha [J, Lippman 2006]
- Added to TiddlyWiki (client-side wiki) → ASciencePad
- ASCIIMath serverside in PHP [Chan 2004], Perl [Nodine 2006]
- Modified to LaTeXMathML.js [Woodall 2006]

Demo of ASciencePad

ASCIIMath is widely used

- Downloaded by thousands of users around the globe
- Integrated into many wikis, blogs, course management systems
- Augmented with ASCIIsvg and a JS scientific calculator [J 2004]
- Merged into WYSIWYG web editors HTMLArea, Xinha [J, Lippman 2006]
- Added to TiddlyWiki (client-side wiki) → ASciencePad
- ASCIIMath serverside in PHP [Chan 2004], Perl [Nodine 2006]
- Modified to LaTeXMathML.js [Woodall 2006]

Demo of ASciencePad

ASCIIMath is widely used

- Downloaded by thousands of users around the globe
- Integrated into many wikis, blogs, course management systems
- Augmented with ASCIISvg and a JS scientific calculator [J 2004]
- Merged into WYSIWYG web editors HTMLArea, Xinha [J, Lippman 2006]
- Added to TiddlyWiki (client-side wiki) → ASciencePad
- ASCIIMath serverside in PHP [Chan 2004], Perl [Nodine 2006]
- Modified to LaTeXMathML.js [Woodall 2006]

Demo of ASciencePad

ASCIIMath is widely used

- Downloaded by thousands of users around the globe
- Integrated into many wikis, blogs, course management systems
- Augmented with ASCIISvg and a JS scientific calculator [J 2004]
- Merged into WYSIWYG web editors HTMLArea, Xinha [J, Lippman 2006]
- Added to TiddlyWiki (client-side wiki) → ASciencePad
 - ASCIIMath serverside in PHP [Chan 2004], Perl [Nodine 2006]
 - Modified to LaTeXMathML.js [Woodall 2006]

Demo of ASciencePad

ASCIIMath is widely used

- Downloaded by thousands of users around the globe
- Integrated into many wikis, blogs, course management systems
- Augmented with ASCIISvg and a JS scientific calculator [J 2004]
- Merged into WYSIWYG web editors HTMLArea, Xinha [J, Lippman 2006]
- Added to TiddlyWiki (client-side wiki) → ASciencePad
- ASCIIMath serverside in PHP [Chan 2004], Perl [Nodine 2006]
- Modified to LaTeXMathML.js [Woodall 2006]

Demo of ASciencePad

ASCIIMath is widely used

- Downloaded by thousands of users around the globe
- Integrated into many wikis, blogs, course management systems
- Augmented with ASCIISvg and a JS scientific calculator [J 2004]
- Merged into WYSIWYG web editors HTMLArea, Xinha [J, Lippman 2006]
- Added to TiddlyWiki (client-side wiki) → ASciencePad
- ASCIIMath serverside in PHP [Chan 2004], Perl [Nodine 2006]
- Modified to LaTeXMathML.js [Woodall 2006]

Demo of ASciencePad

ASCIIMath is widely used

- Downloaded by thousands of users around the globe
- Integrated into many wikis, blogs, course management systems
- Augmented with ASCIISvg and a JS scientific calculator [J 2004]
- Merged into WYSIWYG web editors HTMLArea, Xinha [J, Lippman 2006]
- Added to TiddlyWiki (client-side wiki) → ASciencePad
- ASCIIMath serverside in PHP [Chan 2004], Perl [Nodine 2006]
- Modified to LaTeXMathML.js [Woodall 2006]

Demo of ASciencePad

ASCIIMath is widely used

- Downloaded by thousands of users around the globe
- Integrated into many wikis, blogs, course management systems
- Augmented with ASCIISvg and a JS scientific calculator [J 2004]
- Merged into WYSIWYG web editors HTMLArea, Xinha [J, Lippman 2006]
- Added to TiddlyWiki (client-side wiki) → ASciencePad
- ASCIIMath serverside in PHP [Chan 2004], Perl [Nodine 2006]
- Modified to LaTeXMathML.js [Woodall 2006]

Demo of ASciencePad

Future of ASCIIMathML

- Standardize language as shorthand for a pMathML subset?
- Develop MathML \rightarrow ASCIIMath (reverse) translator
- Expand language to use **Unicode symbols**, i.e. UnicodeMathML.js
- Adapt to the **MS Word 2007** linear formula syntax
- This syntax is quite **similar** to ASCIIMath
- Developed by **Murray Sargent** since the 1970s
- MS Word is widely used, so this will become a **de facto standard**

Future of ASCIIMathML

- Standardize language as shorthand for a pMathML subset?
- Develop MathML \rightarrow ASCIIMath (reverse) translator
- Expand language to use **Unicode symbols**, i.e. UnicodeMathML.js
- Adapt to the **MS Word 2007** linear formula syntax
- This syntax is quite **similar** to ASCIIMath
- Developed by **Murray Sargent** since the 1970s
- MS Word is widely used, so this will become a **de facto standard**

Future of ASCIIMathML

- Standardize language as shorthand for a pMathML subset?
- Develop MathML \rightarrow ASCIIMath (reverse) translator
- Expand language to use **Unicode symbols**, i.e. UnicodeMathML.js
- Adapt to the **MS Word 2007** linear formula syntax
- This syntax is quite **similar** to ASCIIMath
- Developed by **Murray Sargent** since the 1970s
- MS Word is widely used, so this will become a **de facto standard**

Future of ASCIIMathML

- Standardize language as shorthand for a pMathML subset?
- Develop MathML \rightarrow ASCIIMath (reverse) translator
- Expand language to use **Unicode symbols**, i.e. UnicodeMathML.js
 - Adapt to the **MS Word 2007** linear formula syntax
 - This syntax is quite **similar** to ASCIIMath
 - Developed by **Murray Sargent** since the 1970s
 - MS Word is widely used, so this will become a **de facto standard**

Future of ASCIIMathML

- Standardize language as shorthand for a pMathML subset?
- Develop MathML \rightarrow ASCIIMath (reverse) translator
- Expand language to use **Unicode symbols**, i.e. UnicodeMathML.js
- Adapt to the **MS Word 2007** linear formula syntax
 - This syntax is quite **similar** to ASCIIMath
 - Developed by **Murray Sargent** since the 1970s
 - MS Word is widely used, so this will become a **de facto standard**

Future of ASCIIMathML

- Standardize language as shorthand for a pMathML subset?
- Develop MathML \rightarrow ASCIIMath (reverse) translator
- Expand language to use **Unicode symbols**, i.e. UnicodeMathML.js
- Adapt to the **MS Word 2007** linear formula syntax
- This syntax is quite **similar** to ASCIIMath
 - Developed by **Murray Sargent** since the 1970s
 - MS Word is widely used, so this will become a **de facto standard**

Future of ASCIIMathML

- Standardize language as shorthand for a pMathML subset?
- Develop MathML \rightarrow ASCIIMath (reverse) translator
- Expand language to use **Unicode symbols**, i.e. UnicodeMathML.js
- Adapt to the **MS Word 2007** linear formula syntax
- This syntax is quite **similar** to ASCIIMath
- Developed by **Murray Sargent** since the 1970s
- MS Word is widely used, so this will become a **de facto standard**

Future of ASCIIMathML

- Standardize language as shorthand for a pMathML subset?
- Develop MathML \rightarrow ASCIIMath (reverse) translator
- Expand language to use **Unicode symbols**, i.e. UnicodeMathML.js
- Adapt to the **MS Word 2007** linear formula syntax
- This syntax is quite **similar** to ASCIIMath
- Developed by **Murray Sargent** since the 1970s
- MS Word is widely used, so this will become a **de facto standard**

A convenient standard for typing math Unicode?

- Keyboards are “fairly” standard
- Can sit at a computer in Greece or Japan and type an email
- Can handwrite formulas and communicate with nonenglish speakers
- But can't just start a math program and type math
- This is a standardization problem
- Mathematics is a language
- It needs a standard keyboard input format
- LaTeX / Mathematica / Maple / Maxima syntax is not the answer
- Students shouldn't have to learn to type $\sin\pi$ or $\text{Sin}[\text{Pi}]$

A convenient standard for typing math Unicode?

- Keyboards are “fairly” standard
- Can sit at a computer in Greece or Japan and type an email
- Can handwrite formulas and communicate with nonenglish speakers
- But can't just start a math program and type math
- This is a standardization problem
- Mathematics is a language
- It needs a standard keyboard input format
- LaTeX / Mathematica / Maple / Maxima syntax is not the answer
- Students shouldn't have to learn to type `\sin\pi` or `Sin[Pi]`

A convenient standard for typing math Unicode?

- Keyboards are “fairly” standard
- Can sit at a computer in Greece or Japan and type an email
- Can handwrite formulas and communicate with nonenglish speakers
- But can't just start a math program and type math
- This is a standardization problem
- Mathematics is a language
- It needs a standard keyboard input format
- LaTeX / Mathematica / Maple / Maxima syntax is not the answer
- Students shouldn't have to learn to type $\sin\pi$ or $\text{Sin}[\text{Pi}]$

A convenient standard for typing math Unicode?

- Keyboards are “fairly” standard
- Can sit at a computer in Greece or Japan and type an email
- Can handwrite formulas and communicate with nonenglish speakers
- But can't just start a math program and type math
- This is a standardization problem
- Mathematics is a language
- It needs a standard keyboard input format
- LaTeX / Mathematica / Maple / Maxima syntax is not the answer
- Students shouldn't have to learn to type $\sin\pi$ or $\text{Sin}[\text{Pi}]$

A convenient standard for typing math Unicode?

- Keyboards are “fairly” standard
- Can sit at a computer in Greece or Japan and type an email
- Can handwrite formulas and communicate with nonenglish speakers
- But can't just start a math program and type math
- This is a standardization problem
- Mathematics is a language
- It needs a standard keyboard input format
- LaTeX / Mathematica / Maple / Maxima syntax is not the answer
- Students shouldn't have to learn to type `\sin\pi` or `Sin[Pi]`

A convenient standard for typing math Unicode?

- Keyboards are “fairly” standard
- Can sit at a computer in Greece or Japan and type an email
- Can handwrite formulas and communicate with nonenglish speakers
- But can't just start a math program and type math
- This is a standardization problem
- Mathematics is a language
- It needs a standard keyboard input format
- LaTeX / Mathematica / Maple / Maxima syntax is not the answer
- Students shouldn't have to learn to type $\sin\pi$ or $\text{Sin}[\text{Pi}]$

A convenient standard for typing math Unicode?

- Keyboards are “fairly” standard
- Can sit at a computer in Greece or Japan and type an email
- Can handwrite formulas and communicate with nonenglish speakers
- But can't just start a math program and type math
- This is a standardization problem
- Mathematics is a language
 - It needs a standard keyboard input format
 - LaTeX / Mathematica / Maple / Maxima syntax is not the answer
 - Students shouldn't have to learn to type $\sin\pi$ or $\text{Sin}[\text{Pi}]$

A convenient standard for typing math Unicode?

- Keyboards are “fairly” standard
- Can sit at a computer in Greece or Japan and type an email
- Can handwrite formulas and communicate with nonenglish speakers
- But can't just start a math program and type math
- This is a standardization problem
- Mathematics is a language
- It needs a standard keyboard input format
- LaTeX / Mathematica / Maple / Maxima syntax is not the answer
- Students shouldn't have to learn to type $\sin\pi$ or $\text{Sin}[\text{Pi}]$

A convenient standard for typing math Unicode?

- Keyboards are “fairly” standard
- Can sit at a computer in Greece or Japan and type an email
- Can handwrite formulas and communicate with nonenglish speakers
- But can't just start a math program and type math
- This is a standardization problem
- Mathematics is a language
- It needs a standard keyboard input format
- LaTeX / Mathematica / Maple / Maxima syntax is not the answer
- Students shouldn't have to learn to type `\sin\pi` or `Sin[Pi]`

A convenient standard for typing math Unicode?

- Keyboards are “fairly” standard
- Can sit at a computer in Greece or Japan and type an email
- Can handwrite formulas and communicate with nonenglish speakers
- But can't just start a math program and type math
- This is a standardization problem
- Mathematics is a language
- It needs a standard keyboard input format
- LaTeX / Mathematica / Maple / Maxima syntax is not the answer
- Students shouldn't have to learn to type $\sin\pi$ or $\text{Sin}[\text{Pi}]$

Why this is urgent

- **Few** (school/undergrad) students know how to type mathematics
- Most math homework is **handwritten**
- Math tests are often multiple choice (presentation not tested)
- Mathematics seems **oldfashioned** to computer savvy youth
- Mathematics education is affected **negatively**
- Difficult to help students by **email** or **chat**
- Online **interactive** math content is **low**
- Ironically, it's hard to **do** math on a computer!

Why this is urgent

- **Few** (school/undergrad) students know how to type mathematics
- Most math homework is **handwritten**
- Math tests are often multiple choice (presentation not tested)
- Mathematics seems **oldfashioned** to computer savvy youth
- Mathematics education is affected **negatively**
- Difficult to help students by **email** or **chat**
- Online **interactive** math content is **low**
- Ironically, it's hard to **do** math on a computer!

Why this is urgent

- **Few** (school/undergrad) students know how to type mathematics
- Most math homework is **handwritten**
- Math tests are often multiple choice (presentation not tested)
- Mathematics seems **oldfashioned** to computer savvy youth
- Mathematics education is affected **negatively**
- Difficult to help students by **email** or **chat**
- Online **interactive** math content is **low**
- Ironically, it's hard to **do** math on a computer!

Why this is urgent

- **Few** (school/undergrad) students know how to type mathematics
- Most math homework is **handwritten**
- Math tests are often multiple choice (presentation not tested)
- Mathematics seems **oldfashioned** to computer savvy youth
- Mathematics education is affected **negatively**
- Difficult to help students by **email** or **chat**
- Online **interactive** math content is **low**
- Ironically, it's hard to **do** math on a computer!

Why this is urgent

- **Few** (school/undergrad) students know how to type mathematics
- Most math homework is **handwritten**
- Math tests are often multiple choice (presentation not tested)
- Mathematics seems **oldfashioned** to computer savvy youth
- Mathematics education is affected **negatively**
- Difficult to help students by **email** or **chat**
- Online **interactive** math content is **low**
- Ironically, it's hard to **do** math on a computer!

Why this is urgent

- **Few** (school/undergrad) students know how to type mathematics
- Most math homework is **handwritten**
- Math tests are often multiple choice (presentation not tested)
- Mathematics seems **oldfashioned** to computer savvy youth
- Mathematics education is affected **negatively**
- Difficult to help students by **email** or **chat**
- Online **interactive** math content is **low**
- Ironically, it's hard to **do** math on a computer!

Why this is urgent

- **Few** (school/undergrad) students know how to type mathematics
- Most math homework is **handwritten**
- Math tests are often multiple choice (presentation not tested)
- Mathematics seems **oldfashioned** to computer savvy youth
- Mathematics education is affected **negatively**
- Difficult to help students by **email** or **chat**
- Online **interactive** math content is **low**
- Ironically, it's hard to **do** math on a computer!

Why this is urgent

- **Few** (school/undergrad) students know how to type mathematics
- Most math homework is **handwritten**
- Math tests are often multiple choice (presentation not tested)
- Mathematics seems **oldfashioned** to computer savvy youth
- Mathematics education is affected **negatively**
- Difficult to help students by **email** or **chat**
- Online **interactive** math content is **low**
- Ironically, it's hard to **do** math on a computer!

Why this is urgent

- **Few** (school/undergrad) students know how to type mathematics
- Most math homework is **handwritten**
- Math tests are often multiple choice (presentation not tested)
- Mathematics seems **oldfashioned** to computer savvy youth
- Mathematics education is affected **negatively**
- Difficult to help students by **email** or **chat**
- Online **interactive** math content is **low**
- Ironically, it's hard to **do** math on a computer!

Conclusion

- Math uses formulas since they are **short** and **precise**
- Typed linear math notation needs to be **standardized**
- ASCIIMath is **system neutral** and fairly language independent
- Also **easy** to learn, use and implement (on top of MathML)
- Translates into a **well-defined** subset of Presentation MathML
- Matches well with **existing** typed math notations
- **It fills a need for mathematical communication**

Conclusion

- Math uses formulas since they are **short** and **precise**
- Typed linear math notation needs to be **standardized**
- ASCIIMath is **system neutral** and fairly language independent
- Also **easy** to learn, use and implement (on top of MathML)
- Translates into a **well-defined** subset of Presentation MathML
- Matches well with **existing** typed math notations
- **It fills a need for mathematical communication**

Conclusion

- Math uses formulas since they are **short** and **precise**
- Typed linear math notation needs to be **standardized**
- ASCIIMath is **system neutral** and fairly language independent
- Also **easy** to learn, use and implement (on top of MathML)
- Translates into a **well-defined** subset of Presentation MathML
- Matches well with **existing** typed math notations
- It fills a need for **mathematical communication**

Conclusion

- Math uses formulas since they are **short** and **precise**
- Typed linear math notation needs to be **standardized**
- ASCIIMath is **system neutral** and fairly language independent
- Also **easy** to learn, use and implement (on top of MathML)
- Translates into a **well-defined** subset of Presentation MathML
- Matches well with **existing** typed math notations
- It fills a need for **mathematical communication**

Conclusion

- Math uses formulas since they are **short** and **precise**
- Typed linear math notation needs to be **standardized**
- ASCIIMath is **system neutral** and fairly language independent
- Also **easy** to learn, use and implement (on top of MathML)
- Translates into a **well-defined** subset of Presentation MathML
- Matches well with **existing** typed math notations
- It fills a need for **mathematical communication**

Conclusion

- Math uses formulas since they are **short** and **precise**
- Typed linear math notation needs to be **standardized**
- ASCIIMath is **system neutral** and fairly language independent
- Also **easy** to learn, use and implement (on top of MathML)
- Translates into a **well-defined** subset of Presentation MathML
- Matches well with **existing** typed math notations
- It fills a need for **mathematical communication**

Conclusion

- Math uses formulas since they are **short** and **precise**
- Typed linear math notation needs to be **standardized**
- ASCIIMath is **system neutral** and fairly language independent
- Also **easy** to learn, use and implement (on top of MathML)
- Translates into a **well-defined** subset of Presentation MathML
- Matches well with **existing** typed math notations
- It fills a need for mathematical communication

Conclusion

- Math uses formulas since they are **short** and **precise**
- Typed linear math notation needs to be **standardized**
- ASCIIMath is **system neutral** and fairly language independent
- Also **easy** to learn, use and implement (on top of MathML)
- Translates into a **well-defined** subset of Presentation MathML
- Matches well with **existing** typed math notations
- **It fills a need for mathematical communication**

Acknowledgements

Many thanks to the

[Institute for Mathematics and its Applications](#)

[World Wide Web Consortium](#) (HTML, MathML, SVG)

[Design Science](#) (MathPlayer),

[Adobe](#) (SVGview)

[Netscape](#) (JavaScript)

[Mozilla](#) (Firefox)

[SourceForge.net](#) and the [open-source community](#)