Discrete Structures: Overview

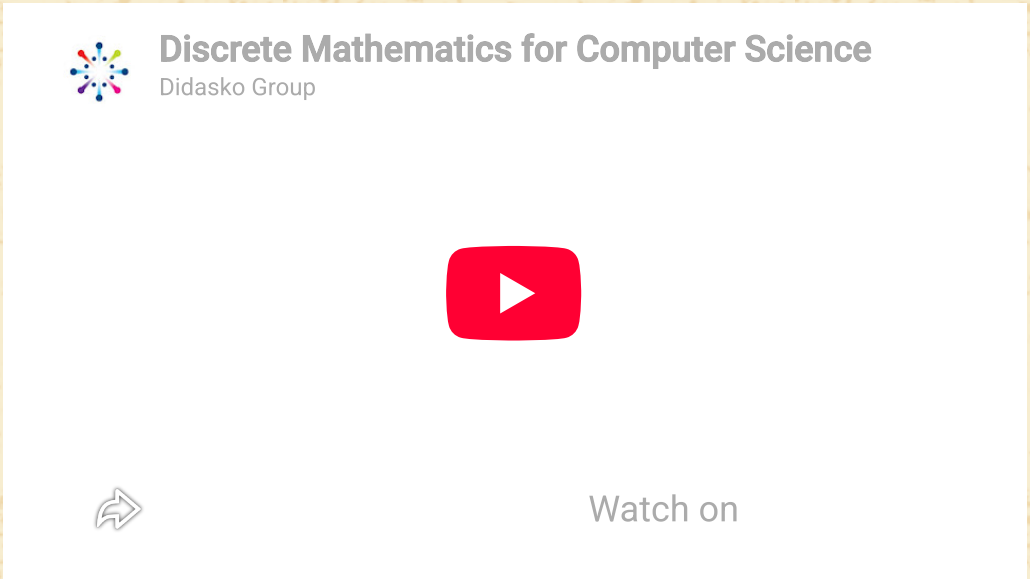
CISC 2210 — CUNY Brooklyn College
Lecture Notes

Definition of Discrete Structures

What are **Discrete Structures**?

Definition of Discrete Structures

The video at https://www.youtube.com/watch?v=q4L-wUF3yig defines Discrete Structures in one way:



https://www.youtube.com/watch?v=q4L-wUF3yig

Definition of Discrete Structures

Fun follow-up questions:

1. What kinds of life problems does the subject of **Discrete Math** deal with?
2. What math is the opposite of Discrete Math, and what life problems does it deal with?
3. Can you recall an example from the video of a Discrete Math problem/challenge?
4. For each of the following life problems, tell if it belongs to Discrete Math or not:
    a. Finding how many students take both CISC 2210 and CISC 3115 this semester.
    b. Finding on which day this year Mars will be the closest to Earth.
    c. Showing that there are 10 integers between 11 and 20, inclusively. (***Bonus question***: Can you list them?)

Definition of Discrete Structures

Discrete Mathematics is the study of mathematical data structures that are discrete rather than continuous, meaning they consist of distinct and separate values rather than forming a continuum (like in calculus, which involves continuous functions and real numbers.)

Since Discrete Mathematics deals with countable, finite, or logically distinct objects, it includes topics like set theory, logic, discrete functions, combinatorics, discrete probability, Boolean algebra, and graph theory.

While some of these topics may seem abstract, they actually form the foundation for computer logic, cryptography, and error detection methods, which you'll study in future CISC classes you take, such as CISC 3310 (Principles of Computer Architecture).

Because Discrete Mathematics underlies the principles of computing, it provides essential tools for algorithm design, data structures, and formal verification techniques. In other words, although programming itself involves writing code, the theoretical foundation provided by this course ensures that the programs you write are efficient (fast and/or light), correct, and adjustable.

Definition of Discrete Structures

Being Computer Scientists, you are taking this course not only because your major requires it (🤪) but also because computational problems often involve discrete rather than continuous values, making this field directly relevant to programming.

Discrete Structures provides the logical reasoning and mathematical tools needed to design and analyze programs. Examples:

- Topics like **Boolean algebra** and **propositional logic** are essential for understanding how circuits and decision-making processes work because computers operate on binary logic.
- Concepts from **graph theory** and **combinatorics** help optimize databases, networks, and search engines.
- For applications like cryptography and cybersecurity, knowledge of **number theory** and **modular arithmetic** is necessary to secure communications and protect data.
- Because algorithms are central to computer science, studying **recurrence relations** and **complexity analysis** allows developers to evaluate the efficiency of different approaches.
- Even in artificial intelligence and machine learning, understanding **probability theory** and **set theory** helps in designing robust models and ensuring correct data representation.

Benefits of Studying Discrete Structures: Summary

- You can make your programs work more efficiently by understanding the math/logic behind the statements or commands in your program.
- You can actually prove whether a program you wrote does its job correctly or not; these techniques are studied in our course!
- Our course is a prerequisite for many other, advanced CISC classes, such as CISC 3310 and CISC 3230.
- Many CS/InfoSys students often take electives offered by the MATH department, such as MATH 2001 and MATH 3107. Learning Discrete Structures will considerably ease your understanding of these subjects!

CISC 2210: Topics

Topics that we'll cover in this course are:

- Topic 1: Discrete Structures: Overview (This Chapter!)
- Topic 2: Sets & Sequences
- Topic 3: Logic & Simple Proof Methods
- Topic 4: Relations
- Topic 5: Functions
- Topic 6: Counting & Combinatorics
- Topic 7: Discrete Probability
- Topic 8: Recurrence & Recursion
- Topic 9: Boolean Algebra
- Topic 10: Matrices
- Topic 11: Graphs & Trees

CISC 2210: Topics

Any questions so far?

Intro to $\LaTeX$

As we've already mentioned on the syllabus, in our course we will also learn a bit of programming in a language called **LaTeX** (pronounced as "late-tech".)

- This is a high-quality typesetting language that is widely used for producing professional-looking documents, particularly those containing mathematical formulas and symbols, scientific notations, and complex formatting.
- It is built on top of **TeX** (pronounced as "tech"), an earlier typesetting language created by Donald Knuth in the late 1970s, and provides structured document preparation features.
- Fun fact: the beautified symbol $\LaTeX$ is written in LaTeX!
  All you need to write in LaTeX code to create this stylized symbol is:

  `\LaTeX`

  anywhere in your source code! The same is true for $\TeX$ and `\TeX`.

Intro to $\LaTeX$

The reasons we study $\LaTeX$ are:

- To eliminate the need to write math symbols and structures by hand: you'll submit all your homework assignments as PDF files created using $\LaTeX$!
- Typesetting in $\LaTeX$ will reduce the waste of paper.
- Unlike word processors like Microsoft Word, $\LaTeX$ separates content from formatting, allowing users to focus on writing while the system handles layout and typesetting. That is, it is quite time-consuming to write math in Word.
- $\LaTeX$, unlike Word, has a rich set of packages that define plenty of useful symbols instantly, without the need to spend hours on finding the right symbol and editing/styling the document.
- LaTeX ensures consistent formatting across large documents, reducing the risk of layout inconsistencies. It is designed to handle complex document formatting, making it ideal for writing research papers, books, theses, and technical reports.
- LaTeX is a popular typesetting language used in many scientific fields, including CS and math, so gaining programming experience in this language is beneficial for a computer scientist.

Compiling and Running $\LaTeX$

No prior knowledge or experience of $\LaTeX$ is required: the rest of these slides will be dedicated to learning all the basic and necessary $\LaTeX$ commands that are needed throughout the semester. Additional $\LaTeX$ commands and statements will be discovered gradually as we keep covering new topics.

pdfLaTeX is a popular type of a compiler that turns your $\LaTeX$ source code into a PDF file. As part of your HW for this course, you will submit the PDF files generated from the $\LaTeX$ code you write (and the source code, too.)

Get $\LaTeX$ up and running on your computer free-of-cost with one or more of the following:

- If you have a Windows, Mac, or Linux computer, installing **MiKTeX**, a free and open-source app, is recommended: https://miktex.org/.
- You may also edit and compile $\LaTeX$ code via your Google Chrome browser using the free $\LaTeX$ Editor extension: https://chromewebstore.google.com/detail/latexeditor/jjbmkdcaffjgaakgpbnhgpjngegpmap.
- **Overleaf** is a free, online website for creating $\LaTeX$-based PDF documents: https://www.overleaf.com/.

Hello $\LaTeX$!

Here is our first full $\LaTeX$ source code that can generate a PDF file (plus a couple of comments):

```
% A one-line comment starts with the % sign.
\documentclass{article}

% LaTeX packages (= libraries) should be imported here via the \usepackage{} command.

\begin{document}
Hello \LaTeX!
\end{document}
```

Fun question: After this program is compiled, what will the resulting PDF file contain/show?

Hello $\LaTeX$!

In the source code on the previous slide:

- The command: `\documentclass{article}` defines the document type, which is an article in this case (can also be `book`, `report`, etc.).
  - Above, `article` is a required argument to the command `\documentclass{}`.
  - Some commands accept no arguments at all (like `\LaTeX`), while others may also accept optional arguments, too. Example: `\documentclass[12pt]{article}` will set the font size of the entire document to 12 points.
- You can create a comment by putting a `%` (percentage sign) before it.
  - A comment can also start in the middle of a line:

    ```
    \documentclass[12pt]{article} % An article with a font size of 12 points.
    ```
- The pair of commands: `\begin{}` and `\end{}` are called an **environment**, which is type of scope in your code. Specifically, `\begin{document}` and `\end{document}` define the document scope: everything between these two will be output to the PDF document.

Coming Up Next...

Next lecture, we'll learn more useful $\LaTeX$ commands and environments!

Coming Up Next...

Any questions so far?

Font settings

The following source code shows how to make font-related changes:

```
\documentclass{article}

% This part (between \documentclass{article} and \begin{document}) is called the preamble of the LaTeX document.
% Here you will (1) import packages whose commands you'll use in the document, (2) define your own functions/commands (which are also called `macros' in LaTeX), and (3) change any settings, such as indentation, margins, etc., which will affect the entire document.


%---------------------------------------

% Imported packages:
```

Font settings

font_style_commands.pdf

Paragraph settings

The following source code shows how to make paragraph-related changes:

```
\documentclass{article}

% Imported packages:
\PassOptionsToPackage{hyphens}{url} % Special hyperref package option that breaks long links
across several lines.
\usepackage{hyperref} % For hyperlinks (e.g., \url{})

%---------------------------------------

% Changing Document Settings:
\setlength{\parskip}{10pt} % Set the space between paragraphs to 10pt.
\setlength\parindent{0pt} % Eliminating paragraph indentation.
```

Paragraph settings

paragraph_style_commands.pdf 1 / 6 111%

1

2

This code example contains commands for changing paragraph-related settings.

Note: For each of the below, there may be more than one way in LaTeX to achieve the shown changes.

To start text on a new line (that is, to begin a paragraph,) leave an empty line between the current and the previous paragraph:

```
This is the 1st paragraph
```

Adding Other Objects

The following source code shows how to add other items to your document:

```
\documentclass{article}

% Imported packages:
\PassOptionsToPackage{hyphens}{url} % Special hyperref package option that breaks long links
across several lines.
\usepackage{hyperref} % For hyperlinks (e.g., \url{})
\usepackage{amsmath} % For the align environment.
\usepackage{graphicx} % For \includegraphics.

%---------------------------------------

% Changing Document Settings:
```

Adding Other Objects

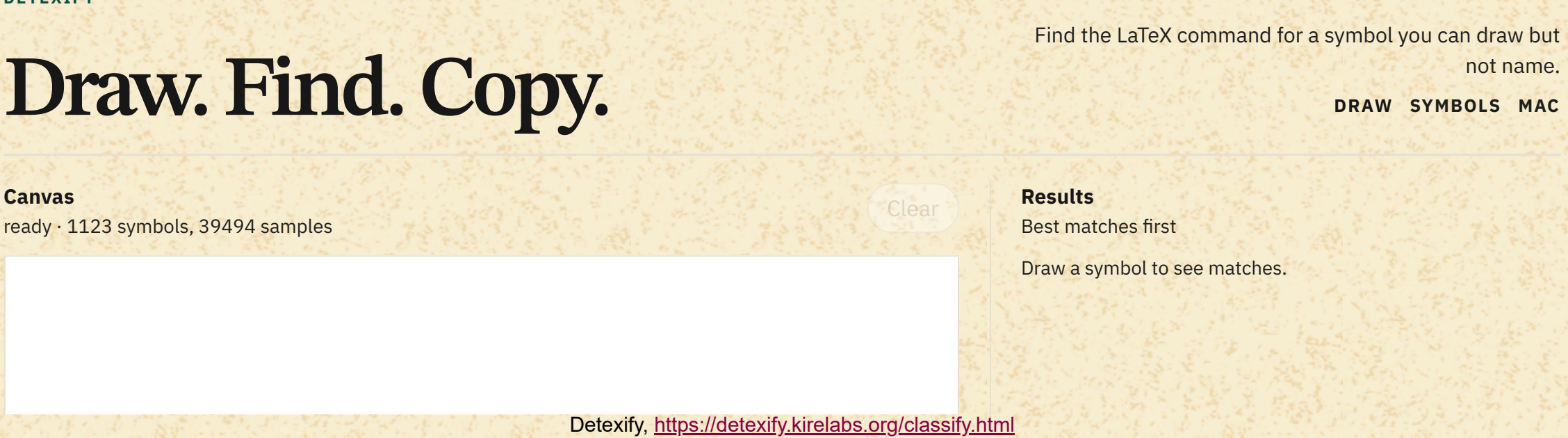
adding_more_objects.pdf

Adding Other Objects

Any questions so far?

$\LaTeX$ Symbol Recognizer: Detexify

The following amazing web app will translate your drawing of a $\LaTeX$ symbol into similar $\LaTeX$ commands!

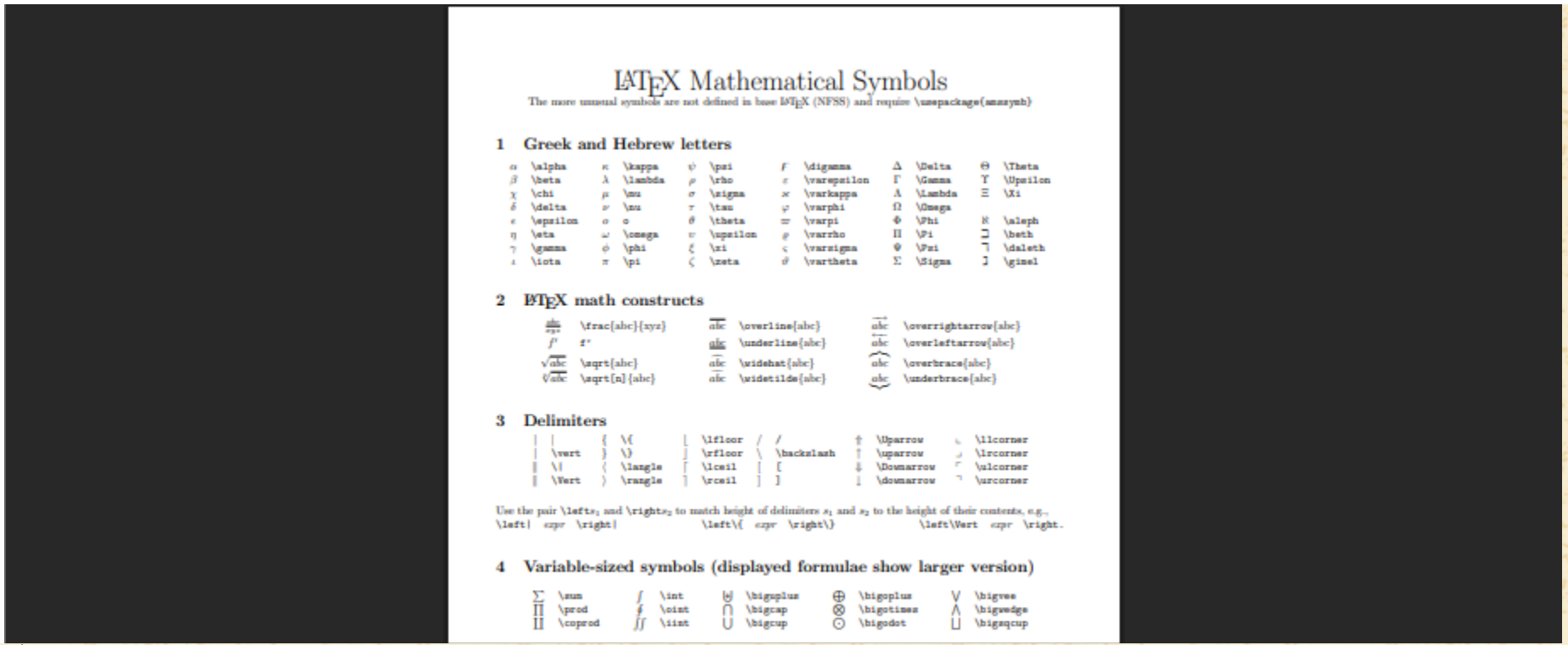


Detexify, https://detexify.kirelabs.org/classify.html

Common LaTeX Math Symbols: List

symbols.pdf 1 / 4 114%

1



LaTeX Mathematical Symbols

The more unusual symbols are not defined in base LaTeX (NFSS) and require \usepackage{amssymb}

1 Greek and Hebrew letters

α	\alpha	κ	\kappa	ψ	\psi	$\digamma$	\digamma	Δ	\Delta	Θ	\Theta
β	\beta	λ	\lambda	ρ	\rho	ε	\varepsilon	Γ	\Gamma	Υ	\Upsilon
χ	\chi	μ	\mu	σ	\sigma	$\varkappa$	\varkappa	Λ	\Lambda	Ξ	\Xi
δ	\delta	ν	\nu	τ	\tau	φ	\varphi	Ω	\Omega		
ϵ	\epsilon	o	o	θ	\theta	ϖ	\varpi	Φ	\Phi	$\aleph$	\aleph
η	\eta	ω	\omega	υ	\upsilon	ϱ	\varrho	Π	\Pi	$\beth$	\beth
γ	\gamma	ϕ	\phi	ξ	\xi	ς	\varsigma	Ψ	\Psi	$\daleth$	\daleth
ι	\iota	π	\pi	ζ	\zeta	ϑ	\vartheta	Σ	\Sigma	$\gimel$	\gimel

2 LaTeX math constructs

$\frac{abc}{xyz}$	\frac{abc}{xyz}	$\overline{abc}$	\overline{abc}	$\overrightarrow{abc}$	\overrightarrow{abc}
f'	f'	$\underline{abc}$	\underline{abc}	$\overleftarrow{abc}$	\overleftarrow{abc}
$\sqrt{abc}$	\sqrt{abc}	$\widehat{abc}$	\widehat{abc}	$\overbrace{abc}$	\overbrace{abc}
$\sqrt[n]{abc}$	\sqrt[n]{abc}	$\widetilde{abc}$	\widetilde{abc}	$\underbrace{abc}$	\underbrace{abc}

3 Delimiters

$\vert$	\vert	$\{$	\{	$\lfloor$	\lfloor	$/$	/	$\Uparrow$	\Uparrow	$\llcorner$	\llcorner
$\vert$	\vert	$\}$	\}	$\rfloor$	\rfloor	$\backslash$	\backslash	$\uparrow$	\uparrow	$\lrcorner$	\lrcorner
$\Vert$	\Vert	$\langle$	\langle	$\lceil$	\lceil	$[$	[	$\Downarrow$	\Downarrow	$\ulcorner$	\ulcorner
$\Vert$	\Vert	$\rangle$	\rangle	$\rceil$	\rceil	$]$	]	$\downarrow$	\downarrow	$\urcorner$	\urcorner

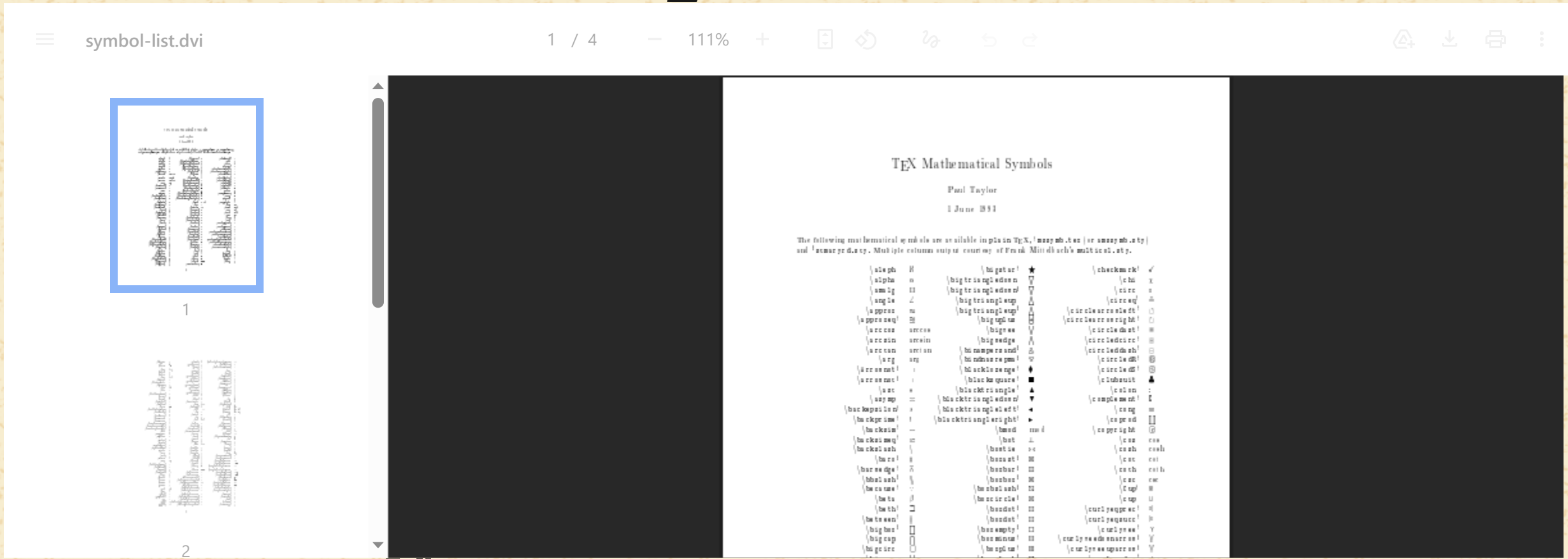
Use the pair \left$s_1$ and \right$s_2$ to match height of delimiters s_1 and s_2 to the height of their contents, e.g.,
\left| *expr* \right| \left\{ *expr* \right\} \left\Vert *expr* \right.

4 Variable-sized symbols (displayed formulae show larger version)

$\sum$	\sum	$\int$	\int	$\biguplus$	\biguplus	$\bigoplus$	\bigoplus	$\bigvee$	\bigvee		
$\prod$	\prod	$\oint$	\oint	$\bigcap$	\bigcap	$\bigotimes$	\bigotimes	$\bigwedge$	\bigwedge		
$\coprod$	\coprod	$\iint$	\iint	$\bigcup$	\bigcup	$\bigodot$	\bigodot	$\bigsqcup$	\bigsqcup		

LaTeX Mathematical Symbols, https://www.cmor-faculty.rice.edu/~heinken/latex/symbols.pdf

Common TEX Math Symbols: List



TEX Mathematical Symbols, https://www.cmor-faculty.rice.edu/~heinken/latex/symbol-list.pdf

More $\LaTeX$ Resources

- [Tutorial] A LaTeX tutorial for academics and non-academics - Practical Typesetting with TeX Live: https://firexfly.com/practical-typesetting/
- [Guide/Tutorial] LaTeX guide for beginners: https://ptmartins.info/latex/
- [Guide/Tutorial] A Minimalist Introduction to LaTeX: https://www.geneseo.edu/~baldwin/reference/MinimalLatex/min_latex.php
- [Guide/Tutorial] The Not So Short Introduction to LATEX 2ε: https://tug.ctan.org/info/lshort/english/lshort.pdf
- [Guide/Tutorial] Learn LaTeX in Y minutes: https://learnxinyminutes.com/docs/latex/
- [Guide/Tutorial] Workshop - Introduction to LaTeX: https://eps-libraries-berkeley.github.io/volt/LaTeX/latex_workshop.html
- [Guide/Tutorial] LaTeX on the Web: https://www.malinc.se/math/latex/latexontheweben.php

---

- [YouTube Tutorial video; 90 min] Just Enough LaTeX to Survive - Full LaTeX Tutorial Series Compilation: https://www.youtube.com/watch?v=lRD_7PSiRAc
- [YouTube Tutorial video; 58 min] Learn LaTeX for Beginners - Full Course: https://www.youtube.com/watch?v=p5Yr0dkrKG8

More LaTeX Resources

Any questions?

Discrete Structures: Overview: Sources

Aspnes, James. (2022). Chapter 1. *Notes on Discrete Mathematics*. License: CC BY-SA: Attribution-ShareAlike.

Davis, Stephen. (2019). Chapter 1. *A Cool Brisk Walk Through Discrete Mathematics*, version 2.2. License: CC BY-SA: Attribution-ShareAlike.

Jamaloodee, Mohamed, et al. Chapter 2. *Discrete Math*. Georgia Gwinnett College. License: CC BY-NC: Attribution-NonCommercial.

Levin, Oscar. (2023). Chapter 0, Section 1 and Appendix C. *Discrete Mathematics: An Open Introduction*, https://discrete.openmathbooks.org/dmoi3/, 3rd edition. License: CC BY-SA: Attribution-ShareAlike.

Sassolas, Mathieu. (2021). Foreword. *Introduction to Discrete Mathematics: An OER for MA-471*, https://academicworks.cuny.edu/qb_oers/179/. License: CC BY-NC: Attribution-NonCommercial.