A STRUCTURED WIKIPEDIA FOR MATHEMATICS
Mathematics in a Web 2.0 World

Henry Lin*
Institute for Theoretical Computer Science, FIT Building 1-208, Tsinghua University, Beijing, China, 100084
henrylin@gmail.com

Keywords: Online Collaboration, Mathematics, Organization, and Web 2.0 Technologies.

Abstract: In this paper, we propose a new idea for developing a collaborative online system for storing mathematical work similar to Wikipedia, but much more suitable for storing mathematical results and concepts. The main idea proposed in this paper is to design a system that would allow users to store mathematics in a structured manner, which would make related work easier to find. The proposed system would have users use indentation to add a hierarchical structure to mathematical results and concepts entered into the system. The hierarchical structure provided by the indentation of results and concepts would provide users with additional search functionality useful for finding related work. Additionally, the system would automatically link related results by using the structure provided by users, and also provide other useful functionality. The system would be flexible in terms of letting users decide how much structure to add to each mathematical result or concept to ensure that contributors are not overly burdened with having to add too much structure to each result. The system proposed in this paper serves as a starting point for discussion on new ideas to organize mathematical results and concepts, and many open questions remain for new research.

1 INTRODUCTION

As the amount of research in the mathematical sciences continues to grow, it is becoming increasingly difficult to stay up to date and find research that is relevant to one's line of work. A single conference or journal can produce hundreds of pages of mathematical work each year, and many disciplines have multiple conferences and journals devoted to them occurring each year. Moreover, as different research areas continue to expand and become more interconnected, a researcher working on some topics may have to stay familiar with research from many different subject areas and disciplines. For example, a theoretical computer scientist working on a topic like algorithmic game theory may have to stay up to date on papers appearing in conferences and journals in theoretical computer science, operations research, economics, and mathematics. As a result of the large amount of new research produced every year, mathematical results are often forgotten or overlooked, only to be rediscovered later with much additional effort.

To address the challenge of making related mathematical work easier to find, in this paper, we propose developing a new online system, which would store mathematical results in a simple structured manner and would make related work easier to find. Our new system would be a collaborative website like Wikipedia, but it would have additional structure and functionality to make finding related work easier. In order for our system to be useful, we would want our system to be simple to understand and easy to use, yet still have the functionality to link related results automatically, and provide additional search capabilities to find related work.

In line with the goal of making the system easy to use and understand, we simply plan to ask users to add structure to their results by indenting their results in a natural way, forming a hierarchical structure (see Figure 1 for a concrete example). The indenta-

*This work was supported in part by the National Natural Science Foundation of China Grant 60553001, and the National Basic Research Program of China Grant 2007CB807900,2007CB807901.

Figure 1: An example of how indentation would be used to add structure to each result.

A special case of Chernoff's bound

- Given Conditions:
  - Let $x_1, x_2, \ldots, x_n$ be
    - independent random variables
    - binary random variables
      - where each x_i variable has probability $\frac{1}{2}$ of being 0 or 1
  - Let $X = \sum_{i=1}^{n} x_i$
  - Let $\mu = \mathbb{E}[X]$
- Conclusion:
  - $\mathbb{P}[X \geq (1-\delta)\mu] \leq e^{-\delta^2\mu}$, for $\delta \in (0,1)$

tion would be done so that if a line of text is indented further below another line of text, it would mean that the later, more indented line of text describes or modifies the less indented line of text, closest above it. For example in Figure 1, we can see that the line containing "binary random variables" modifies and refines the less indented line above it defining the variables $x_1, x_2, \ldots, x_n$. The additional structure provided by users in this indented manner could allow for some very useful search functionality, and by analyzing the text and indentation structure of each mathematical result, the system could identify and link related results by looking for similar text and indentation structure in different results.

Before we describe our proposed system in more detail, we first describe some existing resources on the Internet in Section 2. Then, in Sections 3 and 4, we describe how our system might automatically link related theorems and mathematical objects like complexity classes and algorithmic problems in computer science. In Section 5, we describe some additional search functionality that could be implemented to find related work in our system, and finally, we conclude with some open problems in Section 6.

2 EXISTING MATHEMATICS RESOURCES ONLINE

Fortunately, for researchers working in the mathematical sciences, there are a variety of resources available online containing a large amount of mathematical work. Unfortunately, the problem is that finding related work in many of the existing resources on the Internet can still be a challenge. For example, Wikipedia (Wales et al., 2010), PlanetMath (Egge et al., 2010), and Wolfram's Mathworld (Weisstein et al., 2010) all contain a great deal of work on mathematics, but finding the precise result one is looking for can still be a challenge. The mathematical results in these systems are typically listed alphabetically by name, are assigned keywords to facilitate searching, and/or are organized into broad categories. However, many theorems often have arbitrary names based on the mathematician(s) who discovered them (e.g., the Cook-Levin theorem or Hoeffding's inequality), which can make it difficult to search for a particular theorem, if one does not know the name of the mathematician who discovered it. Furthermore, attempting to find a result by category or keyword search can require a researcher to browse through a large number of results.

Similar problems exist for other mathematical resources on the web, such as the Open Problem Garden (DeVos et al., 2010), which stores open problems in mathematics, and the Complexity Zoo (Aaronson et al., 2010), which stores complexity classes in theoretical computer science. For similar reasons, it can also be difficult to search for related algorithmic problems in the Complexity Garden (Monroe et al., 2010) and the NP Compendium (Crescenzi et al., 2010), which store results on various algorithmic problems in theoretical computer science. (We use 2010 for the citation year because many of the above resources are still being developed and improved upon, although many were initially created earlier).

The only resource on the web (as far as the author knows) that does a very good job of organizing related results appears to be the Scheduling Zoo (Brucker and Knust, 2010), which stores results known about various scheduling problems. In order to search for results known about a particular scheduling problem, a user is allowed to select various parameters and conditions which define a scheduling problem, and then query the system to see if anything is known about the problem selected. The only limitation of this resource is that it was built specifically to store results related to job scheduling problems. It currently cannot be used to store other types of results, and outside users cannot contribute new knowledge to the system. In contrast, the system we propose to develop would seek to match the organization and search capability as provided by the Scheduling Zoo, but it would be capable of storing a wider variety of results. Moreover, any user would be allowed to contribute to it.

It is our hope that our new system would help organize and link related mathematical results and mathematical objects provided in the resources mentioned above. Our resource would not necessarily replace the resources mentioned above, but would

complement them. For example, our system might store and link related mathematical theorems, but it might not list any proofs if they are already provided by other websites. For those proofs, our system might just provide links to proofs listed on existing resources, like Wikipedia and Planetmath. Similarly, our system might only store and link the definitions of related complexity classes, and provide links to relevant entries in the Complexity Zoo for users to find out more about those complexity classes on the Complexity Zoo website itself.

3 ORGANIZING MATHEMATICAL THEOREMS

In this section, we illustrate how our system might work to link related mathematical theorems by imagining what would happen as related theorems are entered into the system. As we will see, our system will have three main mechanisms for linking related results. The first mechanism simply creates a list of related work for each entry in the system by scanning entries for similar structure and text. Results with the most similarity in terms of structure and text, with the result being examined, would be displayed highest on the list of related results for the entry being examined. Furthermore, if any two results are similar enough, the system may also opt to display both results on the same page, or otherwise, it may also create a drop down box so that users can traverse directly between two related results. We will illustrate all three of these mechanisms with some examples below.

Let us first imagine that our first theorem shown in Figure 1 is added into the system, and then the second theorem shown in Figure 2 is also added to the system. Assuming that this second theorem is entered into the system as shown in Figure 2, note that this theorem has exactly the same indentation structure and text as the first theorem in Figure 1, except for the last line in the conclusion. As our system automatically searches for related results, it would look at each existing entry in the system, and check how closely its structure and text matches the theorem being examined. Indeed, if the two results were entered as shown, it would be easy for the system to detect that these two results were related and list them highly in each other's related results list.

Although one might wonder if we can really expect a second user to enter this second theorem in exactly the same format as in the first theorem, this might not be too hard to imagine, if we assume that the second user first tried to search the system for results similar to his/her result before entering his/her new result. By using a standard keyword search to look for results containing the words "independent," "binary," and "random variable" it might not be unreasonable to assume that our second user would have found our first theorem. Assuming that he was able to find our first result shown in Figure 1, it would not be hard to imagine that this second user would just copy the text used to describe the first result, and only change the conclusion line. In this manner, both results would be closely linked as described. In the worst case, where two users created two very different entries which were not similar at first, we would hope that some user would discover this later and help put both entries in a similar format, which would link them automatically.

Figure 2: A second theorem added to our system.

Another special case of Chernoff's bound

- Given Conditions:
  - Let $x_1, x_2, \ldots, x_n$ be
    - independent random variables
    - binary random variables
      - where each random variable x_i has probability $\frac{1}{2}$ of being 0 or 1
  - Let $X = \sum_{i=1}^{n} x_i$
  - Let $\mu = \mathbb{E}[X]$
- Conclusion:
  - $\mathbb{P}[X \geq (1+\delta)\mu] \leq e^{-\delta^2\mu}$, for $\delta > 0$

Additionally, note that the two results have exactly the same text and structure listed in terms of their "Given Conditions." For convenience, we might imagine that our system could be designed to automatically list these two results together on the same page, so that one does not have to navigate between pages to learn about these two very related results. The special conditions that would cause two results to be automatically listed on the same page might have to be fixed and specifically implemented by the system designer, but in case the defined rules do not give the best results, we would also add functionality to allow users to decide for themselves whether or not two results should be placed on the same page, or be placed on two separate pages.

Now, as we imagine more results being added to the system, we might also imagine that the theorem shown in Figure 3 would be closely linked to the first two theorems as well, if added to the system. This new theorem only differs from the first two theorems in terms of the conclusion and in terms of the line that defines the probability with which each random vari-

Figure 3: A third theorem added to our system.

A general case of Chernoff's bound

- Given Conditions:
  - Let $x_1, x_2, \ldots, x_n$ be
    - independent random variables
    - binary random variables
      - *where each x_i variable has probability p_i of being 1, and probability $(1-p_i)$ of being 0, for $i = 1, \ldots, n$*
  - Let $X = \sum_{i=1}^{n} x_i$
  - Let $\mu = \mathbb{E}[X]$
- Conclusion:
  - $\mathbb{P}[X \geq (1+\delta)\mu] \leq (e^{-\delta}/(1+\delta)^{1+\delta})^{\mu}$, for $\delta > 0$

able is 0 or 1 (shown in italics), so this result would also be listed highly among the related results of the first two theorems. Also, since this result only differs from the previous two results at one point in the "Given Conditions" section, we might imagine that our system would take this opportunity to link these results with a drop down box at their point of difference. For example, the third theorem would have a drop down box to switch the condition that defines the random variable x_i to have probability p_i of being 1, to a condition that defines each x_i variable to have probability 1/2 of being 1. Some thought might be needed to decide when the system should automatically link two or more related results with a drop down box, but in case the defined rules do not give the best results, we would also make sure to give users the power to link and unlink results with a drop down box, if the system does not produce good results.

Finally, we show two more results in Figure 4 that would be listed as results related to our previous three results. The first result shown in Figure 4, Hoeffding's inequality, might also be linked to the first three results with a drop down box because it would only involve switching one line (shown in italics) into two adjacent lines in the previous three theorems. The second result shown in Figure 4, the Azuma-Hoeffding inequality, might not be automatically linked with the prior four results with a drop down box, however, because it may contain too many line differences with the four prior results. It would however most likely still be displayed very highly on the related results list of each of the prior four results, and users would be allowed to manually link these results with a drop down box that switches the relevant lines of text.

By linking these five results as described, we

Figure 4: Two more theorems that would be linked as related results.

Hoeffding's Inequality

- Given Conditions:
  - Let $x_1, x_2, \ldots, x_n$ be
    - independent random variables
    - *such that $x_i \in [a_i, b_i]$ almost surely, for $i = 1, \ldots, n$*
  - Let $X = \sum_{i=1}^{n} x_i$
  - Let $\mu = \mathbb{E}[X]$
- Conclusion:
  - $\mathbb{P}[X \geq \mu + \delta] \leq e^{-2\delta^2 / \sum_{i=1}^{n} (a_i - b_i)^2}$, for $\delta > 0$

Azuma-Hoeffding Inequality

- Given Conditions:
  - Let $x_1, x_2, \ldots, x_n$ be
    - *such that $Y_i = x_1 + \ldots + x_i$ forms a martingale, for $i = 1, \ldots, n$*
    - *such that $x_i < c_i$ almost surely, for $i = 1, \ldots, n$*
  - Let $X = \sum_{i=1}^{n} x_i$
  - Let $\mu = \mathbb{E}[X]$
- Conclusion:
  - $\mathbb{P}[X \geq \delta] \leq e^{-\delta^2 / (2\sum_{i=1}^{n} {c_i}^2)}$, for $\delta > 0$

would hope that users would have a much easier time of finding these related results. If someone was unfamiliar with the Azuma-Hoeffding inequality, but at least knew of one of the first four related results, like the special case of Chernoff's bound, then they could first browse for Chernoff's bound. Then upon reaching the Chernoff bound entry, they could then browse the related works listed to find the Azuma-Hoeffding inequality.

4 ORGANIZING ALGORITHMIC PROBLEMS AND COMPLEXITY CLASSES

In this section, we provide some additional examples on how our system could be used to organize and link related complexity classes and algorithmic problems. These examples show how our system might be used to make it easier to find complexity classes listed in the Complexity Zoo (Aaronson et al., 2010)

Figure 5: An example showing how the vertex cover problem is related to the vertex dominating set problem.

Vertex Cover

- Given Input:
  - A graph $G = (V, E)$
- Required Output:
  - A subset of nodes $V' \subseteq V$ of minimum size such that
    - Each *edge in E* is adjacent to a node in V'

(a) Vertex Cover Problem Definition

Vertex Dominating Set

- Given Input:
  - A graph $G = (V, E)$
- Required Output:
  - A subset of nodes $V' \subseteq V$ of minimum size such that
    - Each *node in V* is adjacent to a node in V'

(b) Vertex Dominating Set Problem Definition

and the algorithmic problems listed in the Complexity Garden (Monroe et al., 2010) and NP Compendium (Crescenzi et al., 2010).

4.1 Using the System to Organize Algorithmic Problems

In Figure 5, we illustrate how the vertex cover problem could be linked to the vertex dominating set problem. Note that the problems are defined in very similar ways, except the vertex cover problem requires that *each edge in E is adjacent to a node in* V', while the vertex dominating set problem requires that *each vertex in V is adjacent to a node in* V'. Our system would recognize that these two problems as very similar and link the two results together by creating a drop down box for the last line of each definition. The drop down box would allow users visiting the vertex cover page for example to switch the last line of the problem definition to require that *each vertex in V is adjacent to a node in* V' instead, and thus allow the user to reach the related problem of finding a minimum vertex dominating set.

Furthermore, it is not hard to see that many other algorithmic problems could be linked in a similar way. For example, the minimum cut, minimum k-cut, minimum multi-cut, and minimum multiway cut all have very similar definitions which could be linked as well. Moreover, each of the problems mentioned above has a variant where the objective is to be maximized, and those versions could also be linked.

4.2 Using the System to Organize Complexity Classes

In Figure 6, we illustrate how the complexity classes NP, RP, BPP, and P can all be linked together. Note that each complexity class has a very similar definition, except for the quantifiers which specify how many computation paths must accept on 'yes' instances, and how many computation paths must reject on 'no' instances. Thus the complexity classes mentioned could all be linked by allowing the user to select how many computation paths must be accepted by a 'yes' instance (at least one, at least 1/2, at least 2/3, or all) and how many computation paths must be rejected by a 'no' instance, by using two drop down boxes. Although each combination of acceptance and rejection probability requirements might not yield a standard well-known complexity class, our system would inform users when they select a combination of requirements, which does not yield a standard complexity class. Note that the complexity classes co-NP and co-RP would also be linked with the above results. (It may not be entirely obvious how to get the system to automatically recognize this situation and create the linked structure described above, but it should not be too hard to have functionality, which would allow users to create the structure described above on their own). Lastly, note that when the entry is set so that all computation paths accept for 'yes' instances and all computation paths reject for 'no' instances, the complexity class we have is equivalent to the complexity class P. Even though this is not the standard way of defining the complexity class P, it may be useful for users to know that the last complexity class defined below is an equivalent definition for the complexity class P.

5 EXTENDED SEARCH FUNCTIONALITY

If we look at the indented structure provided in our examples in Section 3, we see that the structure provided could be useful for providing additional search functionality. For example, if we ask users to structure each theorem with a "Given Conditions" section and a "Conclusion" section, we could make queries to the system that could ask for all theorems that contain

Figure 6: An example showing how the complexity classes NP, RP, BPP, and P all have very similar definitions.

NP: Nondeterministic Polynomial-Time

- The class of decision problems solvable by an NP machine such that:
  - If the answer is 'yes,' *at least one* of the computation paths accept.
  - If the answer is 'no,' *all* of the computation paths reject.

RP: Randomized Polynomial-Time

- The class of decision problems solvable by an NP machine such that:
  - If the answer is 'yes,' *at least* $1/2$ of the computation paths accept.
  - If the answer is 'no,' *all* of the computation paths reject.

BPP: Bounded-Error Probabilistic Polynomial-Time

- The class of decision problems solvable by an NP machine such that:
  - If the answer is 'yes,' *at least* $2/3$ of the computation paths accept.
  - If the answer is 'no,' *at least* $2/3$ of the computation paths reject.

P: Polynomial-Time (Alternate Definition)

- The class of decision problems solvable by an NP machine such that:
  - If the answer is 'yes,' *all* of the computation paths accept.
  - If the answer is 'no,' *all* of the computation paths reject.

certain objects in the "Given Conditions" section or the "Conclusion" section. Moreover, we might allow users to query for certain keywords or indented structures appearing in the "Given Conditions" or "Conclusion" section of a theorem. Similarly when searching for objects like algorithmic problems, we might want to search for all problems which contain a certain keyword like "graph" in the "Given Input" section of the problem, and/or "subset of nodes" in the "Required Output" section of the problem. Other search functionality may also be good to implement, although we leave this as an open question for further thought.

6 CONCLUSION

In this paper, we have presented one idea for developing a system to store mathematical results and concepts in a structured manner, which would serve to help users find related work more easily. However, there are still many open questions that could be asked. For example, the system described in this paper is very simple, but is it too simple? Should other rules be added to ensure that entries are consistent and related concepts are linked properly? How do we handle results which can be written in many different ways? One way would be to add special functionality to allow users to mark various entries as equivalent, and searches would take into account equivalent representations when searching for related results. However, it may become confusing if a result has too many equivalent representations, so it may be good to create some guidelines to ensure that each entry does not have too many redundant and useless representations. Besides the search functionality mentioned above, what other functionality should be implemented so that users can find related work? The system presented here is just one idea for organizing mathematical results and there may be room for refinement and improvement. There may also be other good ideas for storing mathematical results, and we hope that this paper can serve as a starting point for further thought and discussion.

Additionally, there are a few other projects related to designing systems to help mathematicians conduct research. For example, there is the Tricki website (Gowers et al., 2010b), which serves to store common tricks useful for solving mathematical problems, and the polymath project (Gowers et al., 2010a), which seeks to enable many mathematicians to get together to solve the same mathematical problem. These two projects have the same problem of finding good ways to organize and link related proof techniques and ideas for proofs. Can a system be described to help organize techniques for proving mathematical results, and/or organize different ideas for proving a theorem? Lastly, one might ask, are there other tools that could be developed to help researchers find related work or conduct research more effectively in general?

REFERENCES

Aaronson, S., Kuperberg, G., Granade, C., et al. (2010). Complexity zoo. http://qwiki.stanford.edu/wiki/complexity_zoo.

Brucker, P. and Knust, S. (2010). The scheduling zoo. http://www.lix.polytechnique.fr/~durr/query/.

Crescenzi, P., Kann, V., Halldorsson, M., Karpinski, M., and Woeginger, G. (2010). A compendium of NP optimization problems. http://www.csc.kth.se/~viggo/problemlist/.

DeVos, M., Samal, R., et al. (2010). Open problem garden. http://garden.irmacs.sfu.ca.

Egge, N., Krowne, A., et al. (2010). Planetmath. http://www.planetmath.org.

Gowers, T. et al. (2010a). Polymath. http://www.polymathprojects.org.

Gowers, T. et al. (2010b). Tricki. http://www.tricki.org.

Monroe, H. et al. (2010). Complexity garden. http://qwiki.stanford.edu/wiki/complexity_garden.

Wales, J., Sanger, L., et al. (2010). Wikipedia. http://www.wikipedia.org.

Weisstein, E., Wolfram, S., et al. (2010). Mathworld. http://mathworld.wolfram.com/.