

Coordinated Topic Presentations for Information Systems Core Curriculum and Discrete Mathematics Courses

Valerie J. Harvey
harvey@rmu.edu

Computer and Information Systems, Robert Morris University
Moon Township, PA 15108 USA

Peter Y. Wu
wu@rmu.edu

Computer and Information Systems, Robert Morris University
Moon Township, PA 15108 USA

John C. Turчек
turчек@rmu.edu

Computer and Information Systems, Robert Morris University
Moon Township, PA 15108 USA

Herbert E. Longenecker, Jr.
hlongenecker@usouthal.edu

Computer and Information Sciences, University of South Alabama
Mobile, AL 36688 USA

Abstract

This paper provides practical information on how to design and implement discrete mathematics modules for coordinated presentation in core curriculum and discrete mathematics courses and is intended for information systems programs seeking ABET accreditation or already accredited by ABET. These modules reinforce the application relevance of the topics and are selected for core curriculum course suitability and on the basis of needs and interests of IS students and foster motivation and confidence as well as understanding of how the concepts presented serve them in learning and will serve them in career settings. Experiences in the information systems (IS) and information systems management (ISM) programs at Robert Morris University (RMU) guided the design of this paper. IS 2002 Core Curriculum mapping for the RSU program is provided as an example.

Keywords: discrete mathematics, quantitative analysis, ABET, curriculum

1. INTRODUCTION AND RATIONALE

The paper identifies topics that meet instructional needs within the framework of

a discrete mathematics course and also can support core curriculum courses through independent modules. An ABET-accreditable core curriculum is the basis for design of the discrete mathematics course,

as described in Harvey, Wu, and Turchek¹. The IS-2002 mappings of courses are shown,² using the RMU curriculum as an example, so that the examples given in this paper can be related to the IS-2002 model. Available software resources for teaching are cited in each case. This paper is designed to support implementation discrete mathematics in IS programs in a manner that addresses ABET IS accreditation criteria for quantitative analysis.

best ways to assure that ABET accreditation criteria for information systems (IS) programs (ABET, 2003) regarding discrete mathematics were met in its information systems programs for which accreditation was sought. Discrete mathematics requirements are explicitly included in the quantitative analysis specification (ABET IS Standards IV-3, IV-11, IV-13).³

While the relevance of discrete

IS-2002 Courses	RMU BS-IS & BS-ISM Courses
IS 2002.P0 Personal Productivity with IS Technology	INFS1020 Intro Decision Support Systems or INFS2410 Office Info Sys Applications or INFS3470 Decision Support Systems *These are electives
IS 2002.1 Fundamentals of Information Systems	INFS1050 Fund. of Information Systems
IS 2002.2 Electronic Business Strategy, Architecture and Design	INFS3150 Intro Web Dev & E-Comm Techn
IS 2002.3 Information Systems Theory and Practice	INFS3220 Systems Analysis and Design
IS 2002.4 Information Technology Hardware and Systems Software	INFS2210 Operating Systems Concep or INFS2211 Microcomputing Technology (A+)
IS 2002.5 Programming, Data, File and Object structures	INFS2130 Cobol Programming or INFS3140 M Programming or INFS3184 C++ Programming or INFS2120 Visual Basic Programming or INFS3151 Java Programming
IS 2002.6 Networks and Telecommunication	INFS3230 Networks/Data/Computer Comm or INFS3231 Network Technology & Mgt (N+)
IS 2002.7 Analysis and Logical Design	INFS3221 Advanced Sys Analysis/Design
IS 2002.8 Physical design and Implementation with DBMS	INFS4240 Database Management System
IS 2002.9 Physical Design and Implementation in Emerging Environments	INFS2121 Visual Basic Programming II or INFS3130 Advanced Cobol Programming or INFS3141 Adv M & Caché Obj Script Prog or INFS3152 - Adv Java: Application Program or INFS3153 - Adv Java: Applet Programming or INFS3188 - Object-Oriented Applicatn Prog or INFS4150 Adv Web Page Design/Ecomm
IS 2002.10 Project Management and Practice	INFS4810 Project Management

Beginning in 2001, the Robert Morris University (RMU) Computer & Information Systems department began seeking the

mathematics topics has been well-defined for computing in general and for computer science (ACM/IEEE, 2001; ABET computer science Standard IV-11), the needs,

interests, and ambitions of IS students are different from those of computer science students. The general strategy of design for the IS discrete mathematics course has been described in certain publications and presentations.⁴ This paper describes how the ABET discrete mathematics criteria can be met and discrete mathematics can be

2. PROGRAMMING EXAMPLES

In courses mapped to IS 2002.5 and IS 2002.9, certain fundamental topics in logic are vital in programming: (Example 2.1) compound statements (statements joined by logical operators like *AND* or *OR* with the negation operator *NOT*, and conditional statements (if...then). The logical properties of conditional statements and of compound statements involving negation and of the statement equivalences described by DeMorgan's Laws are a source of many programming errors. After the systematic treatment of truth tables and statements, students report more confidence in programming and fewer logic errors. Students particularly find the enhanced understanding of equivalent logical statements of practical use in programming.

3. DATA COMMUNICATIONS AND NETWORKING EXAMPLES

In courses mapped to IS 2002.6, graphs and the logical operator *AND* are of particular interest. Networks are typically modeled as graphs and better understanding of graph concepts is helpful in many parts of teaching networking. (Example 3.1) An interesting example of the *weighted graph* is the model of the Maximum Transmission Unit (MTU), to represent restrictions on the size of packets that may be transmitted through a part of the Internet using a given technology.⁵

(Example 3.2) Weighted graphs play a central role in Open Shortest Path First protocol of the Internet. In networking class this type of graph introduces students to the least-weight path concept.⁶ In the discrete mathematics course, the same graph is used in teaching the general concepts of path and path length.

Students have a strong interest in wireless technologies and cell phones. (Example 3.3) Venn diagrams model overlapping

appropriately presented to information systems students most effectively through a set of coordinated topic presentations that directly link the discrete mathematics course with core curriculum courses. Examples are arranged in this paper according to core curriculum topics.

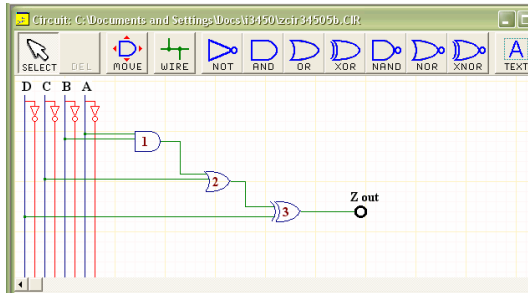
transmission realms and help students comprehend properties of Bluetooth architecture, for example, where a bridge slave may be supported in the intersection of the areas reached by two transmitters, a configuration called a Scatternet, consisting of two Piconets.⁷ A module on graph coloring can be used in both discrete mathematics and networking classes to model the properties of wireless networks requiring that no two adjacent transmitters be assigned the same frequency or chip code pattern. In graph coloring, no two adjacent vertices in a graph may share the same color. The number of different colors needed to support a given graph is called the chromatic number.⁸ Using a teaching tool which automatically assigns colors, students easily and rapidly see that most graphs have a chromatic number of 4 or less. Even in networking class, without the context of additional graph theory instructions, students readily experiment to see what kind of graph can be configured that requires a chromatic number of 5 or more. In the process they discover non-planar embeddings of graphs and speculate about practical applications of these. Students on their own have asked, what about urban areas where a tunnel might constitute a cell, for example, the BART tunnel in San Francisco-Oakland area, or the Liberty tubes in Pittsburgh, or various tunnel in the New York-Newark area. They readily see the importance of such modeling in planning networks.

(Example 3.5) The use of the logical operator *AND* with binary values is used to help understand address masking in the Internet Protocol routing process.

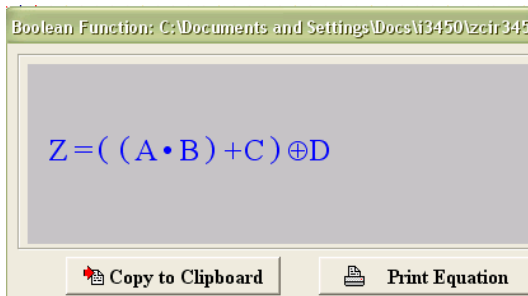
4. HARDWARE AND OPERATING SYSTEM EXAMPLES

Logic gates and the most basic ideas of circuits are covered in textbook for courses in hardware and operating systems courses.⁹ While this material could help students learn about the hardware building blocks of computer technology, it isn't very

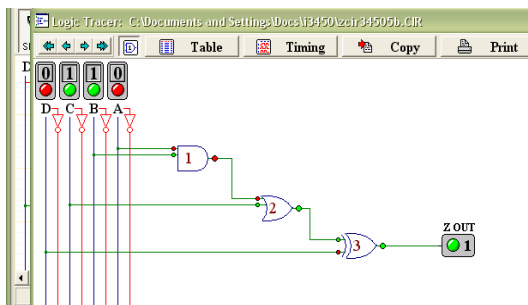
exciting to most students as presented in textbooks.



Hacker (Example 4.1) has authored a teaching tool for developing and testing simple combinatorial circuits.¹⁰ This tool is easy enough to use that it can be presented in IS.2002.4 courses. Experience has shown that students can



develop models with facility and that students' results are professional enough in appearance that some have even requested additional circuit assignments. Through this modeling students gain insights into the hardware implementation of logic. A topic which was abstract and perhaps challenging to approach as presented in textbooks thus becomes enjoyable and productive. The students



have a drag-and-drop interface where they can select appropriate logic gates and utilize them in the design of a model. The model can then be used interactively to test arbitrary inputs. Red and green colors reinforce the 0 and 1 logic values. Students have the option with this tool to display a

table of all the possible input combinations and to see the circuit represented as Boolean equation. If they wish, they can create the circuit as a Boolean equation and review the resulting circuit design. With this tool, students can practice interactively with *AND*, *OR*, *NOT*, *XOR*, and other gates.

State transition diagrams (Example 4.2) are used in operating systems courses to model process management.¹¹ The most common diagrams of this type are quite general. Later in a career environment, students may expect to see the use of this type of diagram to model process management for a particular operating system, such as Unix, Linux, Microsoft Windows, or Solaris.

5. DATABASE EXAMPLES

Discrete mathematics topics in database courses include (Example 5.1) the logic used in SQL, (Example 5.2) SQL queries as logic statements involving predicates, (Example 5.3) SQL aggregates as set partitions, (Example 5.4) the partial orders of creating tables and loading data in database implementation, (Example 5.5) exclusionary queries in SQL, and (Example 5.6) logic and set concepts used in distributed relational database technology.

Logical Operators in SQL (5.1)

As in Example 2.1 above, compound statements involving *AND*, *OR*, *NOT*, also apply to SQL query conditions.

Logical Predicates in SQL (5.2)

When learning to use the database language SQL, students often make errors with regard to predicates. For example, the SQL query for "List the part numbers of all aluminum parts weighing more than 50 kg." It is easy to overlook the fact that the adjective "aluminum" in the specification for the query requires a predicate in SQL, such as, for example, "materialtype = 'aluminum'" A typical SQL query WHERE clause may require a number of predicates. Students who have experience with formal logic are accustomed to interpreting the need to account for predicates and easily avoid such errors.

Aggregates in SQL (5.3)

SQL aggregates in queries using GROUP BY partition sets of database tuples. More

familiarity with the concepts of partitioning sets and with equivalence relations and classes helps students comprehend the impact of GROUP BY in an SQL query.

Partial Orders and Referential Integrity (5.4)

Failure to consider the following can lead to frustrations when students are creating tables and loading data:

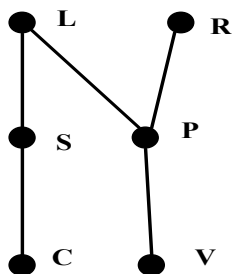
Because of relationships (enforced referential integrity; implemented in relational databases by foreign keys), certain tables are dependent on others. When one table is dependent on another table, that other table must be created or loaded first. One of the most practical capabilities a student can present when seeking employment involving databases, is an understanding of referential integrity and its impact on database design, implementation, and use. Here is the presentation on partial order as given in the discrete mathematics course:

When one table (like **sales_order**) has a foreign key referencing another table (like **customer**), then **customer** must exist before the reference can be created.

When one table (like **sales_order**) has a foreign key referencing another table (like **customer**), then customers must exist (as rows in **customer**) before **sales_order** rows are entered with customer foreign key values.

For example:

if the table **sales_order** is dependent on the table **customer** and contains a foreign key referencing **customer** (like custno),



then the table **customer** must be created/loaded first (before the table **sales_order** is created/loaded)

if the table **sales_order_line** is dependent on the tables **sales_order** and **inventory** and contains foreign key referencing **sales_order** and **inventory** (like sorderno and invno), then both tables **sales_order** and **inventory** must be created/loaded first (before the table **sales_order_line** is created/loaded)

For an entire database, there will be more than one correct order of creating/loading tables with referential integrity enforced according to the dependency structure (this kind of order is called a *partial order* and there are correct variations).

Some general guidelines for determining a correct order of creating/loading:

- First create/load those tables which have no foreign keys (they are not dependent on any other table).
- Then create/load those tables which only require/reference the tables already created/loaded
- Continue this process, following the dependency structures (foreign keys), until all tables are created/loaded.

When tables must be dropped, an inverse order is followed to that described above (tables cannot be removed on which other tables are dependent).

$S = \{\text{customer}(C), \text{sales_order}(S), \text{sales_order_line}(L), \text{product}(P), \text{vendor}(V), \text{restock_order}(R)\}$

$R = \{x, y \text{ is a member of } R \text{ if } x \text{ comes before } y\}$

$R = \{\text{customer}(C) \text{ before } \text{sales_order}(S),$
 $\text{vendor}(V) \text{ before } \text{product}(P),$
 $\text{sales_order}(S) \text{ before } \text{sales_order_line}(L),$
 $\text{product}(P) \text{ before } \text{sales_order_line}(L),$
 $\text{product}(P) \text{ before } \text{restock_order}(R)\}$

A (set of tables in the database) = $\{C, V, P, S, L, R\}$

This Hasse diagram¹² H has two minimal elements, C and V , and two maximal elements, L and R .

The rules derived from this diagram include:

create either the customer table or the vendor table first (minimal elements)

create the customer table before the sales_order table

create both the sales_order and product tables before creating the sales_order_line table

load the customer table before the sales_order table

load both the sales_order and product tables before creating the sales_order_line table

Because of transitivity, create the customer table before the sales_order_line table.

An inverse Hasse diagram H' could be created to represent the order of dropping tables.

The minimal elements of H will be the maximal elements of H' and the maximal elements of H will be the minimal elements of H' .

Rules derived from such an inverted diagram:

drop the sales_order table before the customer table.

drop the sales_order_line table before dropping either the sales_order or product table

drop both the sales_order_line and the restock_order table before dropping the product table.

drop either the sales_order_line or the restock_order table first (minimal elements of H')

As can be seen from this example presentation, some technical detail can be omitted when presenting in the context of a database course.

Exclusionary Queries (5.5)

Instructors in RMU database courses noticed a large proportion of students making mistakes on exclusionary queries (such as "List customer information for every customer who placed an order on a day other than January 10, 2002"). Harvey et al. explored underlying query design and comprehension issues and made practical recommendations on identifying potential sources of error and avoiding incorrect or misleading results.¹³ Examples of such queries from textbooks used in IS instruction include: (1) "Find the customer number, last name, and first name for every customer who did not place an order on October 20, 2003" and (2) "List the order number and order date for every order that was placed by Ferguson's but does not contain an order line for a gas range." and (3) "Show the names and salary of all salespeople who do not have an order with Abernathy Construction..."¹⁴

Logic and Set Theory Used in Distributed Relational Database Technology (5.6)

The logic used in semi-joins, to reduce the transmission costs of distributed queries,¹⁵ and the logic and set concepts used in specifying and implementing vertical and horizontal partitions of relational database tables in distributed environments, offer excellent opportunities to review logic and sets in the discrete mathematics course.

6. PROJECT MANAGEMENT

A particular representation of state transition is important in project management courses mapping to IS 2002.10: the PERT and Gantt Charts (Example 6.1). Students can develop simple or elaborate examples using Microsoft Project. The PERT Chart is based on the critical path concept and exploits the network model.¹⁶ The Internet Center for Management and Business Administration, Inc., supports a concise site covering PERT chart concepts.¹⁷

7. ROLE OF THE PROPOSITIONAL LOGIC TEST (PLT)

Students sometimes overestimate logic capabilities. The Propositional Logic Test (PLT) helps students gain a realistic assessment of their logic capabilities. Necessary support materials for the PLT are available online.¹⁸ Various types of assessment used for the RMU discrete mathematics course are available online.¹⁹

8. INTEGRATION IN THE DISCRETE MATHEMATICS COURSE

Topics which show up in various core curriculum courses may be integrated under a single heading in the discrete mathematics course where students can explore the generality of the concepts and models. Students learn that truth tables support formal logic, programming in Java, Visual Basic, and other languages, and circuits. Logic, a single topic in discrete mathematics supports a range of core curriculum course presentations.

The discrete mathematics course, for example, will provide the general concepts of graph theory so that graph structures can be used with more understanding and confidence in applications topics such as networks. Modules designed for core curriculum courses may have less formal notation than the corresponding modules designed for the discrete mathematics course.

9. CONCLUSION

While not all the details covered in a discrete mathematics course are appropriate for inclusion in a programming course, materials and examples developed for discrete mathematics can be used effectively in teaching programming, networks, operating systems, and many other courses. Since discrete mathematics in an information systems curriculum should directly support the IS core curriculum, the core curriculum courses should be examined (and re-examined) to assure that the proper connection of relevance exists and that students clearly see the applications of the discrete mathematics concepts covered. Examples and exercises can be productively shared by instructors of IS discrete mathematics and the various IS core curriculum courses.

10. REFERENCES

- ¹ Harvey, Valerie J., Wu, Peter Y., and John Turchek, C., "Workshop on Discrete Mathematics for Programs Conforming to ABET Information Systems Accreditation," ISECON, November 4, 2004, Newport, Rhode Island
 - ² J. T. Gorgone et al., IS 2002: Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems" <http://www.aisnet.org/Curriculum/> , ACM, AIS, AITP.
 - ³ *Criteria for Accrediting Information Systems Programs (Effective for Evaluations during the 2002-2003 Accreditation Cycle)*, Computing Accreditation Commission, Accreditation Board for Engineering and Technology, Inc., approved November 3, 2001. The wording on the quantitative analysis requirement is retained in the most recent version of this document for the 2003-2004 accreditation cycle, approved on November 2, 2002.
 - ⁴ Wood, David F., Harvey, Valerie J., and Kohun, Frederick G., "Lifelong Learning: Making Discrete Math Relevant for Information Systems Professionals," *Issues in Information Systems* 2005.
- Longenecker, H E, Daigle, R J and Harvey, V J. "Discrete Mathematics: An Option for ABET Accreditation, but Does it Make Sense as a Support Course for an Information Systems Curriculum?" In *The Proceedings of ISECON 2004*, v 21 (Newport).
- Harvey, Valerie J., Wu, Peter Y., and John Turchek, C., Workshop on Practical Examples for Teaching Discrete Mathematics in an Information Systems Curriculum," AMCIS, August 11, 2005, Omaha, Nebraska
- Harvey, Valerie J. and Holdan, E. Gregory, "Insights from Teaching Discrete

- Mathematics in Information Systems Programs," Report for the Discussion Forum, CoLogNet/Formal Methods Europe Symposium on Teaching Formal Methods (TFM'04), November 19, 2004, Ghent, Belgium.
- Harvey, Valerie J., Wu, Peter Y., and John Turchek, C., "Workshop on Discrete Mathematics for Programs Conforming to ABET Information Systems Accreditation," ISECON, November 4, 2004, Newport, Rhode Island
- Harvey, Valerie J., Harris, Brian, Holdan, E. Gregory, Maxwell, Mark M., and Wood, David F., eds., *Discrete Mathematics Applications for Information Systems Professionals* (Pearson, 2003). Second edition submitted for publication 2005.
- ⁵ Comer, Douglas E., *Computer Networks and Internets with Internet Applications* (Prentice Hall, 2001), p. 332 (§21.5, MTU, Datagram Size, and Encapsulation.)
- ⁶ Comer, Douglas E., *Computer Networks and Internets with Internet Applications* (Prentice Hall, 2001), p. 208 (§13.13)
- ⁷ Tanenbaum, Andrew S., *Computer Networks*, 4th ed. (Prentice Hall, 2003), p. 311 (§4.6.1, Bluetooth Architecture.)
- ⁸ "Colorful Mathematics," Part IV, American Mathematical Society (AMS), "mathematical models for cell phone technology," at: <http://www.ams.org/new-in-math/cover/colorapp5.html> . See also Malkevitch, Joseph, Mathematics and Computing Department, York College (CUNY), Jamaica, New York, "optimal cell configurations, maximizing call capacity of a frequency band," at: <http://www.york.cuny.edu/~malk/tidbits/tidbit-cellphone.html> . An excellent interactive graph coloring exercise is provided by Mawata, Christopher P., University of Tennessee at Chattanooga, Chattanooga, TN, in "Graph Theory Lessons" at <http://www.utc.edu/Faculty/Christopher-Mawata/petersen/>
- ⁹ Burd, Stephen D., *Systems Architecture*, 4th ed. (Thomson Course Technology, 2003), pp. 139-140. An introductory interactive presentation on gates is provided online by Calderwood, Dan, "Introduction to Logic Gates," College of the Redwoods, Eureka, Crescent City, and Fort Bragg, CA at <http://isweb.redwoods.cc.ca.us/INSTRUCT/CalderwoodD/diglogic/index.htm>
- ¹⁰ The software tool, Win Logic Lab, was authored by Charles Hacker of Griffith University and can be downloaded from the Win Logic Lab website at <http://www.gu.edu.au/school/eng/mt/WinLLab.html>; see Charles, and Sitte, Renate, "A Computer-based Interactive Teaching Software for the Tracing of Logic Levels in a Digital Circuit," *Global Journal of Engineering Education* 6, 1 (2002), 85-90. Instruction for RMU students are online at <http://home.earthlink.net/~inforef/i3450circuits.htm> .
- ¹¹ Burd, Stephen D., *Systems Architecture*, 4th ed. (Thomson Course Technology, 2003), p. 430.
- ¹² A Hasse diagram is a directed graph which shows ordering by above-and-below relationships. See Gross, Jonathan L. and Yellen, Jay, *Handbook of Graph Theory* (CRC Press, 2004), p. 150.
- ¹³ Harvey, Valerie J., Baugh, Jeanne M., Johnston, Bruce A., Ruzich, Constance M., and Grant, A. J., "The Challenge of Negation in Searches and Queries," *The Review of Business Information Systems* 7, 4 (Fall 2003): 63-75.
- ¹⁴ (1) and (2) in Pratt, Philip J., *A Guide to SQL*, 6th ed. (Course Technology/Thomson Learning, 2003), p. 106, and (3) in Kroenke, David M., *Database Processing: Fundamentals, Design, and Implementation*, 8th ed. (Prentice Hall, 2002), p. 252.
- ¹⁵ Ricardo, Catherine M., *Databases Illuminated* (Jones and Bartlett, 2004), pp. 192-193.
- ¹⁶ The RMU discrete mathematics introduction to the network model is online at <http://home.earthlink.net/~inforef/i3450networks.htm>

-
- ¹⁷ "PERT," See:
<http://www.netmba.com/operations/project/pert/>
- ¹⁸ See "Teaching Logic in an Illogical World," DIMACS Symposium, at <http://www.cs.cornell.edu/Info/People/gries/symposium/symp.htm> . See Almstrum, Vicki L., "Some Background on the Propositional Logic Test," Department of Computer Sciences, The University of Texas at Austin: <http://www.cs.utexas.edu/users/almstrum/PLT/background.html>
- ¹⁹ Harvey, Valerie J, Wu, Peter Y., and Holdan, E. Gregory, at <http://home.earthlink.net/~inforef/i3450gen.htm> under "Quiz Examples for Study."