Describing the What and Why of Students' Difficulties in Boolean Logic

GEOFFREY L. HERMAN and MICHAEL C. LOUI, University of Illinois at Urbana–Champaign
LISA KACZMARCZYK, Educational Researcher and Consultant
CRAIG ZILLES, University of Illinois at Urbana–Champaign

The ability to reason with formal logic is a foundational skill for computer scientists and computer engineers that scaffolds the abilities to design, debug, and optimize. By interviewing students about their understanding of propositional logic and their ability to translate from English specifications to Boolean expressions, we characterized common misconceptions and novice problem-solving processes of students who had recently completed a digital logic design class. We present these results and discuss their implications for instruction and the development of pedagogical assessment tools known as concept inventories.

Categories and Subject Descriptors: K.3.2 [**Computers and Education**]: Computer and Information Science Education—*Computer science education*; F4.1 [**Mathematical Logic and Formal Languages**]: Mathematical Logic; B.6 [**Logic Design**]:

General Terms: Human Factors, Languages, Verification

Additional Key Words and Phrases: Discrete math, digital logic, formal logic, misconceptions, concept inventory

ACM Reference Format:
Herman, G. L., Loui, M. C., Kaczmarczyk, L., and Zilles, C. 2012. Describing the what and why of students' difficulties in Boolean logic. ACM Trans. Comput. Educ. 12, 1, Article 3 (March 2012), 28 pages.
DOI = 10.1145/2133797.2133800 http://doi.acm.org/10.1145/2133797.2133800

1. INTRODUCTION

Students who are able to reason rigorously with formal logic are more likely to succeed in computer science and engineering [Kim 1995; Owens and Seiler 1996]. When students design software and hardware systems, they must use propositional logic and Boolean algebra (hereafter "Boolean logic") to write specifications, validate designs, test rigorously, and optimize safely. Logic and reasoning skills are typically taught early in computer science and computer engineering curricula (as part of discrete mathematics or digital logic classes) and serve as foundations for many of the classes that follow [IEEE and ACM 2001]. Yet despite the importance of reasoning skills, logic instruction often fails to improve students' reasoning skills or understanding

A shorter, preliminary version of this article appeared at the *ACM International Computing Education Conference* [Herman et al. 2008].
This work was supported by the National Science Foundation under grant DUE-0618589 and DUE-0943318. The opinions, findings, and conclusions do not necessarily reflect the views of the National Science Foundation or the University of Illinois at Urbana-Champaign.
Author's address: G. L. Herman, Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, 1406 W. Green St., Urbana, IL 61801-2918; email: glherman@illinois.edu.

DOI 10.1145/2133797.2133800 http://doi.acm.org/10.1145/2133797.2133800

of formal logic [Cheng et al. 1986; Deer 1969; Nisbett 1993; Stager-Snow 1985], and logic concepts are among the more difficult introductory computer science topics for students to learn [Almstrum 1993, 1996; Goldman et al. 2010].

While it is important and difficult for computer science students to learn formal logic generally, studies with expert instructors in digital logic and computer organization have specifically identified that the sub-task of translating verbal specifications into Boolean expressions is especially important and difficult for digital logic students to master [Goldman et al. 2010]. Based on this previous research, we designed this multi-part study to identify conceptual misunderstandings and poor problem-solving strategies that lead students to fail in these translation tasks. The results of this study can be used directly to improve instruction of discrete mathematics and digital logic classes. In addition, these misconceptions can be used to create conceptual assessment tools.

This study comprises just one portion of a larger study of students' misconceptions that were used to create the digital logic concept inventory (DLCI) [Herman et al. 2009, 2010, 2012, 2011a, 2011c]. A concept inventory (CI) is a multiple-choice test that reliably measures students' conceptual understanding. Knowledge of students' misconceptions allows for the creation of compelling incorrect answers (*distracters*) for a CI. Based on the impact of the Force Concept Inventory (FCI) in physics education [Evans et al. 2003; Hake 1998; Hestenes et al. 1992; Hestenes and Halloun 1995], we believe that well-designed CIs can facilitate the development of better pedagogical approaches as well as promote the adoption of these approaches by providing a way to quantitatively compare the learning gains from different pedagogies. Many in the computer science education community also believe that such rigorous, quantitative assessment tools are needed [Almstrum et al. 2006; Ben-Ari 2005; Clement 2004; McCracken et al. 2001; Tew and Guzdial 2010].

This research was conducted primarily through a traditional qualitative approach based on interviews with students. In these interviews, students were asked to verbalize their thought processes while solving Boolean word problems. We describe the motivation for this approach in Section 2. The four authors coded transcripts from the interviews; our methodology is described in Section 3. We present nine themes that relate to students' difficulties and the thought processes that reinforced these difficulties; these themes are described in Section 4. In Section 5, we discuss some theories about how students develop and reinforce their misconceptions through poor problem solving strategies. We conclude in Section 6 with our ideas for how these findings might be used to modify classroom instruction.

2. BACKGROUND

2.1. Terminology

Definitions of some potentially confusing or contentious terms.

Code. A qualitative research term. A label or tag, usually a word or short phrase, applied to a segment of a transcript that suggests how that segment informs the research objectives and that facilitates the comparison of similarly tagged data [Merriam 2009].

Cue. A perceptual or conceptual trigger that causes a person to retrieve knowledge.

Misconception. The misapplication or mis-cueing of conceptual knowledge within a specific context.

Student. Any student who has taken a digital logic course.

Subject. Any student who participated in the research study.

Theme. A qualitative research term. A synthesis of codes that describes the relationship between multiple codes in a unique way [Merriam 2009].

2.2. Previous Research on Students' Difficulties with Logic

Research has consistently shown that the typical college student does not reason well with formal logic [Petrushka 1984]. For example, Almstrum discovered that students generally perform worse on the logic-based questions on the Computer Science Advanced Placement Exam than on any other type of question on the exam [Almstrum 1993, 1996]. Cheng and Holyoak [1985] found that fewer than 10% of college students reason correctly about the conditional logic statement "if A then B." Selden and Selden confirmed Cheng and Holyoak's findings as they found that fewer than 10% of students could properly apply principles of logic to techniques of proof [Selden and Selden 1995]. Database query research has similarly revealed that students struggle with the ambiguity of Boolean expressions in English when creating or interpreting database queries [Greene et al. 1990; Hoc 1989; Pane and Myers 2000].

To characterize students' misconceptions about various propositional logic operations, researchers at Rutgers University developed the Propositional Logic Test (PLT) [Piburn 1989]. The PLT is a 16-item instrument that tests students' understanding of four logic operations (AND, OR, if-then, and if-and-only-if) through four 4-item subtests. The PLT items are similar to Wason selection tasks [Wason 1966] in that they present the student with a propositional statement, and the student must select which of the four conditions could potentially violate the propositional statement (example Wason tasks can be found in Figures 9 and 10 in Section 4.9). Because the student can select or not select each of the four conditions, each item can be answered in exactly 16 ways where each answer corresponds to one of the 16 binary logic operations. The PLT has been shown to be both reliable and valid [Piburn 1989].

Studies with the PLT have revealed that a student's understanding of logic is correlated with success in science classes [Piburn 1990] and computer science classes [Kim 1995; Owens and Seiler 1996]. Unfortunately, these studies have also revealed that students' scores on the PLT do not increase significantly after instruction on formal logic [Kim 1995; Owens and Seiler 1996]. The PLT revealed that students generally understand the AND concept, but that they struggle to learn OR, if-then, and if-and-only-if [Almstrum 1999]. Almstrum documented that students think of if-then and if-and-only-if as AND, they falsely affirm the consequent or deny the antecedent, and some students think of OR as simply the affirmation of one of the conditions (i.e., A OR $B = B$) [Almstrum 1999]. These results confirm the earlier results of Cheng and Holyoak and Wason. Cheng and Holyoak found that most students mistranslated if-then as AND [Cheng and Holyoak 1985] and Wason found that they committed logical fallacies such as affirming the consequent or denying the antecedent [Wason 1966].

The studies above have revealed that most students struggle with formal logic, but they stop short of examining students' misconceptions about all 16 logical operators and do not offer many suggestions for how to remedy students' misconceptions. Numerous studies have revealed that students' logic misconceptions may originate in the classroom as mathematics and science instructors frequently possess logic misconceptions that they propagate to their students [Goetting 1995; Harel and Sowder 1998; Jungwirth 1985, 1987, 1990]. Other researchers suspect that students learn non-rigorous proof techniques by watching instructors demonstrate difficult concepts through a few examples or counter-examples rather than in-depth proofs [Epp 2003].

Epp proposed that students develop misconceptions about propositional statements, because many propositional statements are open to a variety of interpretations in everyday language [Epp 2003]. In everyday speech, statements are often ambiguous, since there are different acceptable interpretations for these statements in different contexts. For example, “I must go up or down” implies an exclusive-OR even though the verbal construction is an inclusive-OR. In contrast, mathematical language is unambiguous, but instructors often fail to point out how mathematical language is linguistically different from everyday language [Epp 2003]. Consequently, students often fail to discern the subtle differences in the two types of languages.

2.3. How Novices Construct Conceptual Knowledge

Physics education researchers have found that identifying misconceptions alone is not enough to direct instructional reform. In addition, instructors must know why students (commonly referred to as physics novices) make the mistakes they do. Previous research has shown that physics novices think about physics in fundamentally different ways from physics experts [Bransford et al. 1999]. Physics novices focus on surface features of problems rather than on the underlying concepts. They try to recall any formula that seems to match the surface features of the problem they have encountered [Chi et al. 1981; Hardiman et al. 1989]. For example, physics novices focus on the physical objects in a physics problem (e.g., inclined planes, blocks, balls, words like “momentum”), and then they try to recall any formulas that match the variables and objects of the problem. However, physics experts focus on the principles that can be used (e.g., conservation of energy, work) and then use these principles to guide their use of strategies and formulas [Bransford et al. 1999].

There are two diverging theories about the nature of a novice’s conceptual knowledge. One theory argues that a novice’s misconceptions are coherent and consistent [Carey 1999; Vosniadou and Brewer 1992], but the other argues that a novice’s knowledge is fragmented and unpredictable [Clement 1982; diSessa et al. 2004; Wollman 1983]. This second theory has been supported within the computer science and engineering community [Clement et al. 1980; Perkins 1999; Perkins and Martin 1986], and the creators of the FCI used this theory to develop the FCI [Hestenes and Halloun 1995; Hestenes et al. 1992]. According to this theory, a misconception is defined primarily by the contextual cues that lead the novice to misapply pieces of their knowledge. For example, a novice may correctly believe that a rock falls faster than a piece of paper in earth’s atmosphere, but this belief can lead to a misconception when this piece of knowledge is incorrectly cued to explain why a crumpled piece of paper falls faster than an uncrumpled piece of paper. A misconception is the result of improper cueing of knowledge pieces and ad hoc synthesis of these pieces [diSessa et al. 2004; Perkins 1999]. Anderson et al. emphasize that the “training on the cues that signal the relevance of an available skill may deserve much more emphasis than they now typically receive in instruction [Anderson et al. 1998]. In accordance with this second theory, we define a misconception to be the cueing of improper knowledge for a given context rather than a coherent, incorrect theory.

To further explain, it may be useful to imagine that each student possesses several pieces of knowledge. The likelihood that a novice might access each piece of knowledge is context dependent. For instance, a student might cue a causal understanding of “if-then” 80% of the time when thinking about food, but only 20% of the time when thinking about driving laws. A student’s score on a CI estimates the likelihood that a student will cue the correct conceptual knowledge across a variety of contexts. Therefore, an expert who has been trained to cue the correct knowledge will score higher than a novice. When the CI is used to assess the effectiveness of a pedagogy, an instructor can use each item as an estimate of the likelihood that students cue the correct conceptual knowledge in a given context. These likelihoods can indicate which contexts cause students to reveal their misconceptions and how instructors can help their students learn how to cue proper conceptual knowledge.

2.4. Choice of Methodology

In our investigations of student misconceptions in Boolean logic, we seek to describe both the what and the why of their misconceptions. Because most previous studies have explored the "what" of students' misconceptions and few have explored the "why," we took a grounded theory approach to understand why students develop misconceptions in Boolean logic. Grounded theory is a qualitative research method that can be used when little is known about what people do and how they think in a given context. In grounded theory, no theory or hypothesis should be formed prior to the collection of data [Glaser and Strauss 1967]. Instead, theories should emerge from the use of open-ended data collection, and the analysis of this data should inform future data collection [Strauss and Corbin 1998].

Our previous research revealed that Boolean logic is important and difficult for students to learn [Goldman et al. 2010]. Based on this finding, we conducted this study where we asked subjects to think aloud about what they were doing within a familiar context [Ericsson and Simon 1984; Weiss 1994]. After our initial analysis of these interviews, we refined our interview protocol to investigate the questions that emerged from the first round of analysis. These interviews were analyzed with rigorous coding schemes that protect against bias and the premature formation of conclusions. To enhance rigor in the coding schemes, we coded data individually. Then we discussed these codes and emergent themes and retained only those codes and themes on which we agreed unanimously [Strauss and Corbin 1998].

3. METHODOLOGY

Subjects in this study were interviewed for one hour about their understanding of a wide range of concepts in digital logic design. Due to time constraints, each participant was interviewed on only a portion of the selected concepts. The interview questions resembled problems that the subjects may have encountered previously in a digital logic class. Subjects were paid for their participation and gave written consent to be interviewed under IRB approval (University of Illinois at Urbana-Champaign number 07026).

3.1. Subjects

In Spring 2008, seven undergraduate volunteers from the University of Illinois at Urbana-Champaign were interviewed individually about translating English specifications to Boolean expressions. Two women and five men were interviewed individually; two were international students. During a second round of interviews in Spring 2009, 10 more undergraduate volunteers were interviewed. All volunteers were traditional age (18-22) undergraduates majoring in computer science, electrical engineering, or computer engineering who had completed a three-credit, digital-logic design class with a simulation lab in the Fall 2007 or Fall 2008 semesters and had earned grades of B or C (1.7 to 3.3 on a 4.0 scale). Students who had earned B and C grades were selected because their understanding was likely to be less complete, and they were more likely to have misconceptions than students who had earned A grades (greater than 3.3). Our pilot interviews, which are not included in this study, confirmed these expectations, as the interviews with students who had earned grades of A had yielded fewer mistakes or misconceptions.

3.2. Interview Process

Interviews were conducted in a “think-aloud” format: subjects were instructed to vocalize their thoughts as they solved problems and responded to questions [Ericsson and Simon 1984]. Prior to the interview, subjects were briefed on the study’s goal of understanding how they think through various topics in digital logic design. They were told to not expect feedback about whether their answers were correct during the interviews but to expect frequent requests to expand on what they were doing [Ericsson and Simon 1984].

All interviews were recorded using a document camera (which recorded what the subject wrote) and microphone. The audio tracks of the interview recordings were transcribed verbatim, the subjects’ gestures were included in the transcript, and every piece of paper the subject wrote on was scanned electronically.

3.3. Interview Questions

During the first round of this study, the interviews consisted of a set of non-overlapping questions to probe students’ ability to translate a variety of English statements into Boolean expressions. For the second round of this study, the interviews were structured to refine our understanding of where students struggled and of what caused students to manifest misconceptions. We chose to ask the same questions multiple times, but changed the context or the presentation of the problem.

Example of changed context. Subjects were asked to complete (1) a Wason task (see Figure 9) using an unfamiliar context (shapes and numbers) and (2) a Wason task using a familiar context (drinking ages in a bar) (see Figure 10).

Example of changed presentation. Subjects were asked to interpret an English statement by (1) translating the statement directly to a Boolean expression, (2) translating the statement to a truth table, and (3) matching the statement with illustrations of the conditions described by the statement. Subjects had to solve every problem using presentation style (1) and they also needed to solve every problem using presentation style (2) or (3) but not both. We did not ask subjects to solve a problem in all three styles, because we did not want them to think that they were being tricked or misled by the problems.

By asking students to solve the same problem with different presentation styles and contexts, we hoped to answer three questions.

(1) Are student misconceptions consistent, or do they vary based on the task?
(2) Does failing to enumerate all possible cases of a logical statement induce student errors? Alternatively, when students are forced to enumerate all possible cases, do they demonstrate better conceptual knowledge?
(3) Do students have difficulty interpreting English statements correctly, or do they have misconceptions about the nature of Boolean variables? Alternatively, do students misconceive of Boolean variables in general (i.e., struggle to fill-in or interpret truth tables), or do they misconceive only about certain concepts (i.e., only make mistakes concerning specific concepts — NAND, implication, etc.)?

3.3.1. First Round Questions. During first round interviews, all subjects were asked the three questions about Boolean word problems in Figure 1. Questions 2 and 3 were sometimes asked in the opposite order to reduce the impact that answering one question may have had on answering the other question. Question 1 was designed to probe

Question 1. Explain the meaning of *if A then B* in Boolean logic. Give an example that demonstrates your understanding.

Question 2. A campus sandwich shop has the following rules for making a good sandwich:
(1) A sandwich must have at least one type of meat,
(2) A sandwich must have roast beef or ham, but not both,
(3) If a sandwich has turkey then it must also have cheese.

Write a Boolean expression for the allowed combinations of sandwich ingredients using the following variables:
c = 1 when cheese is present, h = 1 when ham is present,
r = 1 when roast beef is present, t = 1 when turkey is present.

Question 3. A recipe for apple pie has the following instructions:
(1) Do not use both allspice and nutmeg simultaneously; and
(2) Use nutmeg if and only if you use cinnamon.

Write a Boolean expression for the allowed combinations of spices for the apple pie using the following variables:
a = 1 when allspice is used, c = 1 when cinnamon is used, n = 1 when nutmeg is used.

Fig. 1. List of first-round interview questions.

Question (1) At minimum the subject should be able to correctly translate *if A then B* to the Boolean expression $\bar{A} + B$.
Identifying the conditional as implication was also expected.

An example that a student might give to demonstrate a deeper understanding of *if A then B* would be, "If a greeting card is blue (A), then it must be a birthday card (B)." This statement is falsified only when the card is blue, but is not a birthday card ($A\bar{B} = \overline{(\bar{A} + B)}$). A blue birthday card would be acceptable (AB); a red birthday card would be acceptable ($\bar{A}B$); and a white wedding card would be acceptable ($\bar{A}\bar{B}$).

Questions (2) & (3) approach: It was expected that the subjects would solve the written Boolean word problems by first translating each rule of the problem statement into a Boolean expression independently. They would then compose the separate expressions into a single expression by ANDing the rules together.

Question (2) solution
rule 1 = $r + h + t$; rule 2 = $r\bar{h} + \bar{r}h = r \oplus h$; rule 3 = $\bar{t} + c$;
Final rule = $(r + h + t) \cdot (r\bar{h} + \bar{r}h) \cdot (\bar{t} + c)$ = Simplified rule = $(r\bar{h} + \bar{r}h) \cdot (\bar{t} + c)$

Although not required, this rule can be simplified by recognizing that rule 2 subsumes rule 1.

Question (3) solution
rule 1 = $\overline{(an)}$ = $a\ NAND\ n$; rule 2 = $nc + \bar{n}\bar{c}$ = $\overline{(n \oplus c)}$;

Final rule = $\overline{(an)} \cdot (nc + \bar{n}\bar{c})$

Fig. 2. List of acceptable answers for first-round interview questions.

the students' conceptual understanding of if-then, while Questions 2 and 3 were designed to simulate questions the students may have encountered in their digital logic design class. A list of acceptable answers to these questions is in Figure 2. Additional clarifying questions were asked in response to what subjects said.

Patrons 1, 2, 3, and 4 each have the following preferences for their sandwiches.
Use the variables, b = 1 when bacon is present; l = 1 when lettuce is present; t = 1 when tomato is present.

Patron 1 prefers a sandwich with bacon by itself.
If a sandwich has tomato, then **Patron 2** prefers a sandwich that also has lettuce.
Patron 3 prefers a sandwich that does not have both lettuce and bacon.
Patron 4 prefers a sandwich that has neither lettuce nor tomato.

Presentation (1)*:* Translate each preference into a Boolean expression that specifies all sandwiches that satisfy the patron's preference.

Presentation (3): Identify the sandwiches that each patron would prefer. Use the paper with pictures of sandwiches.

Fig. 3. Second-round interview questions that used presentation styles (1) and (3).

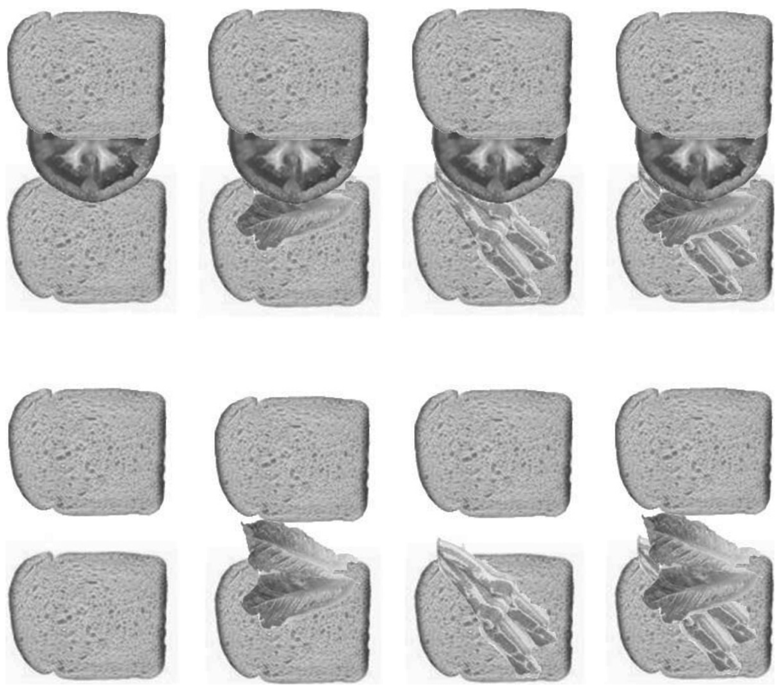


Fig. 4. Images used when subjects were asked to match illustrations with the conditions described by an English statement.

3.3.2. Second Round Questions. During the second round interviews, subjects were asked the questions shown in Figures 3, 4, 5, 7, 9 (see Section 4.9), and 10 (see Section 4.9).

3.4. Data Analysis

The interviews were analyzed using the following steps of grounded theory and qualitative data analysis as described by Kvale [Kvale 1996], Strauss and Corbin [Strauss and Corbin 1998], and Miles and Huberman [Miles and Huberman 1984].

Alex, Beth, and Chris want to order a single, large pizza that they all will like to eat.
Use the variables, p = 1 when the pizza has pepperoni; s = 1 when the pizza has sausage; o = 1 when the pizza has olives.

> Alex will eat pizzas with olives if and only if the pizza also has pepperoni.
> Beth will eat only pizzas that have pepperoni without sausage.
> Chris will eat only pizzas that have exactly two ingredients.

Presentation (1): Write a single Boolean expression that specifies what pizzas they can order.

Presentation (2): Fill out the truth table below to show what pizzas the group can order.

p	s	o	
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Fig. 5. Second-round interview questions that used presentation styles (1) and (2).

Patron 1's preference $= b\bar{l}\bar{t}$
Patron 2's preference $= \bar{t} + l,$
Patron 3 preference $= \overline{lb}$
Patron 4's preference $= \bar{l}\bar{t}$

Alex's preference $= \bar{o}\bar{p} + op$
Beth's preference $= p\bar{s}$
Chris's preference $= ps\bar{o} + p\bar{s}o + \bar{p}so$
Combined preference= $p\bar{s}o$

p	s	o	Alex	Beth	Chris	Combined
0	0	0	1	0	0	0
0	0	1	0	0	0	0
0	1	0	1	0	0	0
0	1	1	0	0	1	0
1	0	0	0	1	0	0
1	0	1	1	1	1	1
1	1	0	0	0	1	0
1	1	1	1	0	0	0

Fig. 6. List of acceptable answers for second-round interview questions.

The four authors analyzed the data: the interviewer (Herman), a former instructor of a digital logic class (Loui), a colleague with content knowledge in Boolean logic (Zilles), and a researcher with extensive experience in qualitative research methods (Kaczmarczyk).

Fill out a truth table for the following function

$$f(x,y,z) = x(y + (\bar{y} + x)z) + y\bar{z}$$

x	y	z	
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Fig. 7. Second-round interview question about truth tables.

Step (1). To avoid bias, all interviews were analyzed. While the interviews included questions on other topics, only the subjects' responses to the Boolean word problems were analyzed, because this study focused on Boolean logic misconceptions.

Step (2). All researchers analyzed the interviews independently without a predetermined coding scheme, as prescribed by grounded theory [Strauss and Corbin 1998]. Principles of grounded theory were used to uncover the subjects' misconceptions, because this paradigm allows the misconceptions to emerge from the data without an a priori theoretical framework that would influence the observations. Eschewing an initial coding scheme also allows for fuller descriptions of what the subjects did correctly or incorrectly for each statement.

Step (3). The four researchers met and discussed every annotation and observation that they had made. To ensure the accuracy and completeness of our coding, a unanimous decision was needed for an annotation to be included for coding or rejected from coding. If a unanimous decision was reached, then it was counted as an agreement; otherwise it was counted as a disagreement. Preliminary code names and definitions were created for every accepted annotation.

Step (4). After all interviews were discussed, the preliminary code names and definitions were refined by two researchers (Herman and Kaczmarczyk) to facilitate the identification of thematic patterns. The refined list of codes and definitions was given to all four researchers to identify the thematic elements of the codes independently. All researchers then met again to discuss the thematic elements that they had noted. A unanimous decision about the presence of a theme was needed for it to be included in the final list of themes. Once we have developed a full list of themes, we can synthesize these themes to develop theories of students' difficulties and misconceptions about Boolean logic.

This process was repeated for the second round of interviews, but one author (Kaczmarczyk) was unavailable.

An inter-rater reliability of 95% was calculated as follows:

$$R = A_n/(A_n + D_n), \tag{1}$$

where R is inter-rater reliability, A_n is the total number of agreements, and D_n is the total number of disagreements.

The conception codes are of most direct interest to this research, because they indicate the misconceptions that can be used to create the concept inventory. These codes also help to gauge the relative difficulty of different concepts. While the action codes will not be used to create the concept inventory directly, these codes offer additional insights about how the subjects developed these misconceptions. Therefore, they provide guidance towards instructional interventions to help students overcome their misconceptions. These codes also demonstrate the expertise level of the subjects.

4. THEMES

The data analysis revealed nine primary themes about how students access or cue their knowledge. The themes were (1) misconceptions about the use of design tools such as Karnaugh maps, (2) a tendency to reduce difficult Boolean operators to easy operators, (3) a tendency to reduce unfamiliar tasks to familiar tasks, (4) if-then translation misconceptions, (5) confusion about the meaning of a false antecedent, (6) treating complemented variables as non-essential, and (7) misconceptions stemming from interference from exposure to a concept in multiple contexts, (8) proof by incomplete enumeration and non-systematic approaches to problem solving, and (9) over-dependence on problem presentation.

4.1. Design Tool Misconceptions

Subjects were proficient with manipulating the basic tools of Boolean logic. All subjects could correctly fill in a truth table when given a complex Boolean expression such as the one in Figure 7. All subjects could derive a Boolean expression for a given truth table. Subjects could even simplify Boolean expressions during this derivation.

Despite a demonstrated proficiency with basic Boolean logic tools, subjects verbally expressed a dislike for using them. Unless prompted by the interviewer to use these tools, subjects opted to use ad hoc reasoning. Examples of this behavior are provided in Section 4.8.

However, subjects demonstrated that they did not fully understand the purpose of the different tools and representations of Boolean logic. When asked to fill in a truth table based on a logical statement, Subject 10 said that he should fill in a Karnaugh map in order to fill in the truth table.

> INTERVIEWER: Can you walk me through how you would fill in this truth table?
> SUBJECT 10: So I could do a Karnaugh map, but I forgot how to do a three variable Karnaugh map so I'm not going to.

This subject did not understand that Karnaugh maps and truth tables are simply alternate forms of the same information and that drawing a Karnaugh map would be redundant. The subject's desire to use the Karnaugh map cannot be explained by the subject's preference, because the subject could not even remember how to create the Karnaugh map. This failure to understand the relationship between Karnaugh maps and truth table confirms previous findings that subjects conceive of design tools as problems in their own right rather than as techniques that they can apply in different contexts [Longino et al. 2006].

4.2. Reduction to Easier Concepts

The coding process indicated that there are two classes of Boolean operators: "easy" and "difficult." Subjects could correctly perform activities involving the easy operators in almost all instances. Subjects showed incomplete understanding of the difficult

Inputs			*Easy operators*					
A	**B**	NAND	XOR	NOR	OR	AND	If-then	If-and-only-if
0	**0**	1	0	1	0	0	1	1
0	**1**	1	1	0	1	0	1	0
1	**0**	1	1	0	1	0	0	0
1	**1**	0	0	0	1	1	1	1

Fig. 8. Truth tables of the NAND, XOR, NOR, OR, AND, if-then, and if-and-only-if concepts. The loops show the similarities between the difficult operators and the easy operators that are troublesome for subjects.

operators: subjects incorrectly reduced them to the easy operators. To facilitate discussion, a set of truth tables for these concepts is given in Figure 8.

4.2.1. AND, OR, NOR, and XOR - Easy Operators. The interviews revealed that AND, OR, NOR, and XOR are easy operators. All subjects consistently demonstrated a correct intuitive knowledge of these concepts and correctly used them in a variety of contexts. In other words, subjects easily cued their knowledge according to principles of logic, even though some of these terms (such as OR) have indefinite meanings in English. Only one subject mistranslated "*A* or *B*, but not both," as OR. All other subjects were able to translate the previous statement correctly and were also able to express the XOR operation using the basic Boolean operators of AND, OR, and NOT. Furthermore, no subject mistranslated the phrase "at least one" from Question 2 Rule 1 (see Figure 1), and many subjects demonstrated a deeper understanding of why OR is the correct translation. We did not find evidence of widespread misconceptions of OR as was found by Almstrum [1999].

4.2.2. If-And-Only-If (XNOR) Reduction. Only three subjects were able to correctly translate the biconditional statement "*A* if and only if *B*." Most subjects reduced the biconditional statement into one of its constituent single-direction conditional statements "If *A* then *B*" or "If *B* then *A*." These same subjects and other subjects also reduced "*A* if and only if *B*" to be "*A* AND *B*". Subjects 1 and 12 reduced "if-and-only-if" to be "if-then" (italics are added for emphasis).

> SUBJECT 12: [When translating olives if and only if pepperoni] *So if it has olives, then it also has to have pepperoni*. But it can also, it doesn't have to have olives [Writes $op + \bar{o}$]. So, the olives and pepperoni, or no olives at all.

Subjects 3, 6, 7, and 11 reduced if-and-only-if to AND, by first reducing "*n* if only if *c*" to "if *c* then *n*" and then reducing "if *c* then *n*" to "*c* AND *n*."

> SUBJECT 7: I interpret [if-and-only-if] as *cinnamon has to be used in order for nutmeg to be used, but not the other way around*. So, ... I guess it's probably [*writes n* ***AND*** *c*]

Two subjects (Subjects 14 and 15) reduced if-and-only-if to AND with no intermediate step. However, both of these subjects later reduced if-and-only-if to if-then when they encountered if-and-only-if in the truth table presentation style (style 2 in Figure 5).

> SUBJECT 14: [Presentation style 1] Cause, by reading the sentence that customers will eat pizza with olives, if and only if the pizza also has pepperoni, [*op*] made the most logical sense to me.

> SUBJECT 14: [Presentation style 2] So if olives is a one, well if olives is a zero, he'll eat it. But if olives is a one, then pepperoni must be a one.

4.2.3. If-Then Reduction. Nine subjects mistranslated implication as "A AND B." Most subjects mistranslated implication as AND only when translating from an English specification to a Boolean expression, but not during the Wason tasks.

> SUBJECT 5: If you have turkey, then you must also have cheese [write $+tc$] so it's **turkey AND cheese**.

4.2.4. Not Both (NAND) Reduction. Several subjects incorrectly reduced the NAND operator to XOR. The phrases "do not use both allspice and nutmeg simultaneously" (see Figure 1) and "sandwich that does not have both lettuce and tomato" (see Figure 3) were mistranslated by more than half of the interview subjects to be XOR expressions.

This misconception could have been created when subjects improperly cued their knowledge on the phrase "not both," because the phrase "not both" appears in both NAND and XOR specifications. The following subject starts with the "not both" phrase and checks only three cases before concluding that he has verified his use of XOR, (the explication of cases is added in italics for clarity).

> INTERVIEWER: So how'd you come up with $\bar{a}c$ **OR** $a\bar{c}$ for "do not use both?"
> SUBJECT 2: Well, when we do not have allspice, I mean it says do not use allspice and nutmeg simultaneously ($\langle a, c\rangle = \langle 1, 1\rangle$), right?
> INTERVIEWER: Okay.
> SUBJECT 2: So if allspice is not being used, we can use cinnamon ($\langle a, c\rangle = \langle 0, 1\rangle$). And if allspice is used, then we cannot use cinnamon ($\langle a, c\rangle = \langle 1, 0\rangle$).

Other subjects mistranslated the "not both" expression into NOR statements. When subjects reduced NAND to NOR, they focused only on the case when $\langle l, b\rangle = \langle 1, 1\rangle$. After verifying that their expression $\bar{l}\bar{b} = 0$ for the $\langle 1, 1\rangle$ case, they concluded that their translated expression was correct.

The two reduction misconceptions (NAND-to-XOR and NAND-to-NOR) provide evidence that subjects retrieved their knowledge by using a few test cases and then cued their knowledge based on the results of these test cases.

4.3. Reduction to Familiar Tasks

There are two classes of Boolean translation tasks: familiar and unfamiliar. Familiar translation tasks require subjects to translate English statements that are directly related to Boolean operations and are therefore commonly covered during instruction.

4.3.1. Reduction to If-Then Translation. In class, subjects were not specifically taught how to translate statements such as "without" or "by-itself". When subjects were translating these statements, they were performing an unfamiliar task. When asked to perform these unfamiliar tasks, subjects exhibited the misconception of reducing concepts to match familiar tasks. These misconceptions appeared when subjects reworded the English statement to be translated before attempting the translation task. For example, Subject 9 rewords the "pepperoni without sausage" requirement into an implication requirement before translating the statement.

> SUBJECT 9: It says Beth will eat pizzas that have pepperoni without sausage. So, if there's no pepperoni, she can eat any pizza out there. ... If there is pepperoni, it has to be not sausage. So, NOT p OR (p AND NOT s) [subject writes: $!p$ OR ($p\&!s$)]

Subjects mostly reduced unfamiliar tasks to the familiar task of translating an if-then statement. Although if-then is a difficult concept (see Sections 4.4 and 4.2), it has a well defined translation that can be memorized, and it is a common statement in both Boolean logic instruction and everyday English.

Presentation style affected the subjects' tendency to reduce the familiarity of tasks. Presentation style 1 affected more reductions to if-then than presentation styles 2 or 3.

4.3.2. Artificially Constraining Problems with the Real World. Subjects also made unfamiliar tasks more familiar by interpreting logical statements according to personal opinion or experience. When given an English statement to translate, the rules and conditions of the created problem do not always match with a subject's perception of the real world. Despite this mismatch, subjects often questioned or validated their answers based on their real-world experience rather than the problem statement.

For example, when asked to determine "how many different possible combinations of ingredients can be placed between two slices of bread," Subject 11 failed to give the empty set case. He later explained, "I don't consider two pieces of bread to be a sandwich! So I'm going to consider that not valid."

4.4. If-Then Misconceptions

Although most subjects recognized the conditional statement, "if A then B," as implication, only three subjects were able to correctly translate or interpret the statement across all contexts. Most subjects demonstrated multiple misconceptions about implication. These misconceptions stemmed from faulty recall (e.g., subjects recalled the expression $\bar{B} + A$ instead of $\bar{A} + B$), reduction to easy operators (See Section 4.2), incomplete case analysis, and struggles with understanding the relationship between the antecedent (A) and the consequent (B).

Two of the subjects who initially mistranslated implication to be "A AND B," later used incomplete case analysis to derive the expression "$AB + \bar{A}B$" after realizing that B could be true by itself without violating implication; both subjects failed to include the $\bar{A}\bar{B}$ case. (Cases are added for clarity.)

> SUBJECT 3: And then [rule] 3 ... I guess would just be like, turkey implies cheese, so let's see ... **turkey AND cheese** [$\langle t, c\rangle = \langle 1, 1\rangle$] because **OR... NOT turkey AND cheese** [$\langle t, c\rangle = \langle 0, 1\rangle$]? I think, because this would be such true, if it has turkey and cheese, but it doesn't say anywhere that cheese cannot be by itself. So this can also be true. [writes $tc + \bar{t}c$].

4.5. False Antecedent Confusion

Half of the subjects demonstrated confusion about the relationship between the antecedent and the consequent in a conditional statement. The most common misconception was that the truth of the antecedent causes the consequent to be true (emphasis added in italics).

> SUBJECT 7: I would say, well, given two statements that are each either true or false you could arbitrarily call one A and the other B and *the only reason why B would be true would be if A is true first. So, if A then B*.

A potentially related misconception is the belief that if the antecedent is false, then the conditional is false.

> SUBJECT 3: You only want to use nutmeg if you use cinnamon. So if I use cinnamon, so cinnamon would be 1, then the use of nutmeg would determine

> the value of this expression. And then *if you don't use cinnamon, then it would automatically be 0, for the whole thing*.

The cause of these misconceptions could stem from interference from the way if-then is used in programming or colloquial English [Epp 2003]. Our observations confirm the previous findings that students generally commit logical fallacies such as affirming the consequent or denying the antecedent [Wason 1966].

Our subjects' struggles with the Wason task reveal a different pattern of misconceptions from those found by Cheng and Holyoak [1966] (see Section 4.9). Our subjects indicated they thought that only the square should be turned over. Cheng and Holyoak do not report this result. During interviews, subjects evaluated only the antecedent and subsequently treated the consequent as an afterthought.

4.6. Non-Essential Complemented Variables

Subjects demonstrated difficulties with using complemented variables. When enumerating cases to evaluate, subjects frequently failed to evaluate the case where all variables were false or the empty set case. During the first round interviews only one subject explicitly evaluated the empty set case for the if-then constructions, and only one subject evaluated the empty set case for the "not both" specification. When subjects failed to check the empty set, they did not find their mistranslations. During the second-round interviews, subjects were forced to check all cases — including the empty set case, because presentation styles 2 and 3 fully enumerated all cases for the subjects. Subjects who initially omitted the empty set or omitted negated variables consistently found their mistakes once forced to fully enumerate all cases.

Subjects also had difficulties including complemented variables in their expressions when encountering the English specifications "by itself," "exactly two," and "without" (see specifications in Figures 3 and 5). While translating the apple recipe into a Boolean expression, subjects translated "allspice by itself" as just a (e.g., $f(a, c, n) = a$) rather than a ANDed with the complements of all other ingredients (e.g., $f(a, c, n) = a\bar{c}\bar{n}$). Similar mistakes were made for the phrase "without." In the following example, the subject incorrectly tranlates their statement of "cinnamon by itself without the nutmeg" as c instead of $c\bar{n}\bar{a}$ and allspice by itself as a instead of $a\bar{c}\bar{n}$.

> SUBJECT 5: You can use cinnamon by itself without the nutmeg, because that doesn't break rule (2) [writes $+c$] ... or you could just use allspice by itself [writes $+a$].

In contrast, when deriving a Boolean expression from a truth table, subjects never omitted negated variables.

4.7. Ambiguity and Interference

Most digital logic students encounter logical constructions in programming contexts and in the context of their everyday use of language. The concepts linked to Boolean operators and expressions have different meanings in those two contexts (e.g., "A or B" in colloquial English is typically an exclusive-OR statement) and the symbols used to represent similar concepts are also assigned different meanings in other contexts (e.g., "$A + B$" means addition in programming, OR in Boolean logic, and sometimes means AND in English shorthand writing).

Because some Boolean operations (e.g., OR, if-then) mean different things in different contexts, we say that these concepts are ambiguous. When subjects borrowed symbols (e.g., using & or + for AND rather than standard symbols $\wedge$ or $\bullet$, or using ! for complementation) from different contexts to represent a concept in the Boolean context, they revealed interference between the contexts of learning. Subjects exhibited misconceptions caused by ambiguity and interference.

The most ambiguous construction is "if-then." On several occasions subjects specifically mentioned that they knew there is a difference between how "if-then" is used in programming or colloquial English and how "if-then" is used in Boolean logic. Despite this knowledge many subjects were unable to articulate the difference between the contexts.

> INTERVIEWER: How would you describe the phrase if A then B in Boolean logic?
> SUBJECT 4: In Boolean logic or in plain English?
> INTERVIEWER: Imagine that you are teaching them.
> SUBJECT 4: Uh ... If A then B would mean that if the expression after the "if," like if you had "if X=1," then some other expression such as X++ or increment X then if X=1 then you would do the statement after that, saying "okay that is true." That's more of a programming statement than it is the Boolean logic approach. The [the digital logic design class] approach would be something more like what it means to have "if-then" is some sort of implication, where if you have one then you have the other.

One subject revealed interference when he never wrote a standard Boolean expression, but wrote statements that resembled programming structures and function calls.

> SUBJECT 9: If olives and pepperoni OR NOT olives [Writes if $(o$ & $p)$ OR $(!o)$].
> SUBJECT 9: So, you could say p AND XOR s and o [Writes: p AND xor(s, o)].

4.8. Proof by Incomplete Enumeration and Other Non-Systematic Approaches

The enumeration of cases to prove the correctness of a logical expression (proof by exhaustion) is a foundational law within Boolean logic, yet subjects often felt they had proved equivalence after enumerating only one or two cases. What we came to refer to as "proof by incomplete enumeration" resulted in two types of errors for the subjects: reduction errors and faulty error correction.

Figure 8 shows that XOR and NAND are equivalent for three cases and that AND, "if-then," and, "if-and-only-if" are equivalent for two cases. Subjects frequently enumerated only the cases where the two concepts were equivalent, and failed to enumerate the cases where the two concepts were not equivalent. For example, Subject 6 checked only one test case ($\langle t, c\rangle = \langle 1, 0\rangle$) for Question 2 Rule 3 (see Figure 1) before deciding his recalled expression was correct. Examples of faulty proofs can be found in Section 4.2.4

> INTERVIEWER: How would you interpret [rule 3] by itself?
> SUBJECT 6 : I would just start with turkey ... okay I think it is t **AND** c.
> INTERVIEWER: And why do you think that?
> SUBJECT 6: Because if it is 1 which means you have **turkey**, and you have 0 **cheese** ($\langle t, c\rangle = \langle 1, 0\rangle$) this statement is 0 which is wrong, and we want this statement to be 1 which means that we want both t **AND** c.

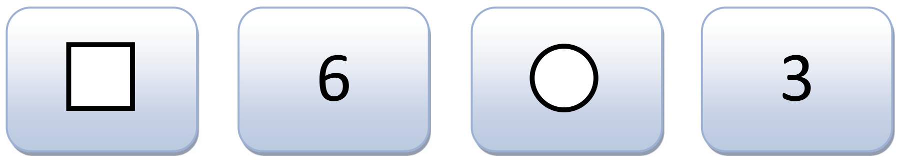
When the structure of the problem presentation forced subjects to fully enumerate all cases (e.g., fill in a truth table), subjects corrected mistakes that they had failed to discover when they used proof by incomplete enumeration.

Although using exhaustive proof techniques helped subjects solve problems correctly, subjects were reluctant to use these proof techniques. Some subjects expressed

Suppose that I have a pack of cards. Each card has a shape drawn on one side and a number written on the other side. Suppose in addition that I claim the following rule is true:

If a card has a square on one side, then it has an odd number on the other side.

Imagine that I now show you these four cards from the pack:



Which card or cards should you turn over in order to decide whether the rule is true or false?

Fig. 9. Second-round interview Wason task question with abstract context.

open dislike for using truth tables and other "brute force" methods. Other subjects tried to use Boolean algebra and identities to solve problems even when asked to fill in a pre-drawn truth table to solve those problems.

4.9. Problem Presentation

While an expert in a field can identify the underlying concepts needed to solve domain specific problems, a novice often relies on the immediately visible features of a problem to determine the solutions to their problems. In this section, we describe which surface features most influenced our subjects' choice of problem-solving strategies.

4.9.1. Context as a Surface Feature.

> SUBJECT 8: [Responding to Figure 9] So if a card has a square, then it has an odd number on the other side. [I have to flip over] the square card. Because, just because we have an odd number, and we flip it over, and there isn't a square there, that's OK. But if we know that we have a square, there better be an odd number on the other side. So it would definitely be this one [puts a check mark on the square card]. That's the only one you need to flip over. Yeah, that's the only one.
> INTERVIEWER: OK, similar question here.
> SUBJECT 8: [Responding to Figure 10] If you're drinking beer, then you need to know if they're over or under 21. If they're drinking Sprite, you really don't care, because anyone can drink that. If you're 25 then you can drink anything, and then if you're 16, you can only drink not an alcoholic product, so you need to get more information about this. So, [I should flip-over] the two end cases [chooses the drinking beer card and the 16 years old card].

A Wason task is a standard cognitive test of formal reasoning. In the Wason-tasks (see Figures 9 and 10), both rules are based on an "If A then B" clause. Both rules can only be violated when $\langle A, B\rangle = \langle 1, 0\rangle$. Therefore, we need to take action (flip the card, request more information) only for those cases where A is true (A) or B is false $(\bar{B})$. The card with the square and the person drinking beer correspond to A, the card with the circle and the person drinking Sprite correspond to $\bar{A}$, the card with the three and the 25 year old correspond to B, and the card with the six and the 16 year old

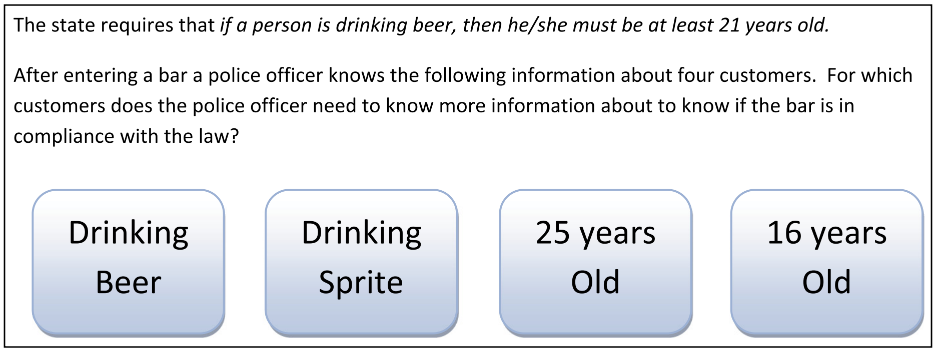


Fig. 10. Second-round interview Wason task question with real-world context.

correspond to $\bar{B}$. Therefore, we must choose both the person drinking beer and the square and the 16 year old and the number six.

Subjects frequently used different reasoning strategies that depended on the surface-level structure, context, or presentation of the problem. For example, Subject 8 ignored all cases that were not directly referenced by the antecedent for the squares and numbers task, but carefully considered both the antecedent and the consequent for the bar task. Similarly, six of 10 subjects solved these two problems with the same set of differing strategies: They incorrectly used only the words in the implication statement (square and odd number) to decide that they needed to turn over only the square and the three cards, but they solved the bar context problem correctly.

In this task, we found that the concrete, everyday context of the second Wason task helped the subjects correctly complete the task. This result disagrees with many of the findings in the mathematics literature, which has shown that everyday contexts are often a hindrance for students when solving algebra word problems [Schoenfeld 1992; Verschaffel et al. 1994].

4.9.2. Words in a Specification as a Surface Feature. Subjects commonly erred during Boolean translation and analysis tasks, because they focused only on cases that were directly implied by the English statement. For example, Subject 16 initially translated if-and-only-if correctly. Once questioned, though, the subject placed extra emphasis on the visible cases (olives and pepperoni as present) and changed the translation to match.

> SUBJECT 16: [translating olives if-and-only-if pepperoni] Well, he'll eat any pizza that has both olives and pepperoni or no olives and no pepperoni.
> INTERVIEWER: How does that make sense to you from the specification?
> SUBJECT 16: Well, it needs to be olives and pepperoni. Wait, [op] seems like a better answer.
> INTERVIEWER: And why do you say that?
> SUBJECT 16: Because [the specification] doesn't say anything about no olives no pepperoni situations. So it needs to be olives AND pepperoni.

Similar emphasis on visible cases were made by subjects who used "proof by incomplete enumeration" to "prove" that "not both" should be translated as XOR or NOR. The case that $\langle A, B\rangle = \langle 1, 1\rangle$ is false is the only case explicitly described by the statement "not both."

Over-reliance on surface features can also be seen in subjects' translation of the expression "if A then B." The statement "if A then B" provides two explicit cases to evaluate, $\langle A, B\rangle = \langle 1, 1\rangle$ and $\langle 1, 0\rangle$, but it implicitly offers information about the $\langle 0, 0\rangle$ and $\langle 0, 1\rangle$ cases. Many subjects failed to address these implicit cases in their spoken reasoning after reducing if-then to AND.

4.9.3. Complete Enumeration Tools as a Conceptual Crutch. When presented with logical English statements to translate, subjects were asked to (1) translate the English statement into a Boolean expression and either (2) use the English statement to fill in a truth table or (3) use the English statement to select illustrations that satisfy the statement. When subjects performed the first translation task, they made mistakes such as omitting negated variables (see Section 4.6), incomplete enumeration (see Section 4.8), and reduction to easier conceptions and familiar tasks (see Sections 4.2 and 4.3). When completing tasks (2) or (3), these mistakes mostly disappeared.

5. DISCUSSION

In this section, we discuss how students cue their conceptual knowledge and consequently how they create their misconceptions. We present a model that may help instructors to predict which types of problems are more likely to cause their students to improperly cue their knowledge. We also describe the limitations of this study.

5.1. Students' Misconceptions and How They Are Created

From our nine themes, we propose answers to our three research questions.

(1) Are student misconceptions consistent, or do they vary based on the task?
(2) Does failing to enumerate all possible cases of a logical statement induce student errors? Alternatively, when students are forced to enumerate all possible cases, do they demonstrate better conceptual knowledge?
(3) Do students have difficulty interpreting English statements correctly, or do they have misconceptions about the nature of Boolean variables? Alternatively, do students misconceive of Boolean variables in general (i.e., struggle to fill in or interpret truth tables), or do they misconceive only about certain concepts (i.e., only make mistakes concerning specific concepts—NAND, implication, etc.)?

Question 1. Our themes revealed that our subjects' conceptions were built upon chaotic and ambiguous conceptual frameworks. Our subjects encountered the words and concepts of Boolean logic in multiple contexts, and these contexts created conceptual interference. Consequently, the subjects' misconceptions manifested differently depending on various contextual cues such as problem presentation or the "real world" context of the problem. The subsections below explain in detail how the different contexts affected how the subjects cued their knowledge.

Question 2. Subjects were reluctant to use, and expressed dislike toward, exhaustive enumeration techniques. When subjects frequently failed to use exhaustive enumeration techniques, they revealed many misconceptions about the Boolean operations and Boolean variables. In contrast, when they were constrained to use exhaustive enumeration techniques, they rarely revealed these same misconceptions. The additional problem structure changed the way that students cued their knowledge and facilitated their use of correct conceptual knowledge.

Question 3. Most subjects could correctly interpret all of the English specifications (except for if-and-only-if) in at least one context or presentation style. Subjects did not make any mistakes when manipulating Boolean variables. These results imply that students' translation errors were not simply due to a language barrier or a wholesale misunderstanding of the specifications. Therefore, students' mistakes when translating from the context of English specifications directly to the context of Boolean expressions seem to be caused by misconceptions that are cued by the presentation style or context of the problems. Likewise, our subjects' belief that complemented variables are non-essential was also context dependent.

5.1.1. Cueing of NAND Misconceptions and the Problems with XOR. The OR and XOR concepts are commonly taught by explaining the difference between the inclusive- and the exclusive-OR: XOR does not include both, but OR does include both. We found compelling evidence that students easily develop a strong association between XOR and the "not both" part of its definition. This strong association can cause students to improperly cue their knowledge of the XOR concept when NAND should be cued.

The "not both" distinction between OR and XOR is especially important when translating from English to Boolean expressions, and becomes a cue for knowledge retrieval. Consequently, many subjects initially retrieved XOR when they encountered the "not both" cue. Because most subjects corrected their initial use of this faulty cue when explicitly required to check all cases, we believe that the added problem structure provides additional cueing information that helped the subjects use the correct interpretation.

In a recent article, VanDeGrift et al. [2010] report that they found that 90% of students correctly interpret the "not both" specification found in Figure 1 prior to formal logic instruction. We believe that these students performed well for two reasons: The context of their study tested students who possessed different cues and the presentation style of their study facilitated proper reasoning techniques. First, if students have not received formal logic training, they are unlikely to have thought deeply about the distinction between inclusive-OR and exclusive-OR and thus are less likely to cue their knowledge of the exclusive-OR based on the phrase "not both;" the primary difficulty of our subjects. Second, VandeGrift et al.'s survey constrains students to examine the validity of the rule for a predefined set of cases and not to translate the English specification into a Boolean expression. Consequently, their students were constrained to use methods akin to the exhaustive enumeration techniques that helped our subjects translate the specification correctly.

5.1.2. Cueing of If-Then. If-then possesses many meanings in colloquial English and programming, but only one of the definitions is used as the technical definition of if-then in logic. The ambiguity and interference of if-then means that there are many possible cues for students to use. We have identified four primary cues for students' retrieval of knowledge: the logic cue, causality cue, the programming cue, and the visual cue. The logic cue leads the student to retrieve the correct logic conception. The causality cue leads the student to retrieve the belief that if-then expresses a causal or AND relationship. The programming cue leads the student to retrieve the belief that if-then statements are evaluated only by evaluating the truth of the antecedent. The visual cue leads the student to retrieve visual or grammatical knowledge that is tangential to the problem. The abundance of cues explains the diversity of misconceptions that students possess and why students succeed more on some tasks than others.

— In the Wason task bar context in Figure 10, the colloquial use of English matches the technical logic definition. Consequently, the context provided the proper logic cue for subjects to correctly solve the problem.
— In the Boolean translation tasks, subjects were presented with two cues: a colloquial cue (the causality cue) from the English statement and a logic cue. The logic cue is less explicit in the "direct to Boolean" translation task than in the "translate to a truth table first" translation task, because the truth table displays all the cases that

Table I. List of Knowledge Cues for If-Then

If-then cue	Information retrieved
Logic cue	Correct Boolean logic formulation of if-then (if p then q is $\bar{p}+q$)
Causality cue	The antecedent causes the consequent to be true (if p then q is p AND q)
Programming cue	Only the truth of the antecedent needs to be checked
Visual cue	Ad hoc reasoning based on visible features of the problem

need to be evaluated. Consequently, when translating to a truth table first, subjects were more likely to use the logic cue to generate their answer.

— In the Wason task shapes and numbers context in Figure 9, the task is different from any problem that the subjects had previously solved and the context did not indicate which cue to use. Consequently, the subjects' answers can be explained by all four cues, and subjects likely cued their knowledge with whichever cue they were most comfortable. Since the subjects had been solving logic problems, the logic cue was readily available and the students successfully completed the Wason task more often than the typical college student [Cheng and Holyoak 1985; Selden and Selden 1995]. It is uncertain if these same subjects would perform as well if they were given the Wason task in a programming class or any non-logic class. Because the causality cue is a generally familiar cue, subjects may have defaulted to this cue and interpreted the if-then rule as AND. Similarly, since many of the students had recently or were concurrently taking a programming class, the programming cue would have been particularly strong for these subjects. Finally, subjects may have frequently chosen to turn over the square and the three, because of a simple pattern matching cue: the statement mentions a square and an odd number and the images provide matching images. Without a familiar method for interpreting the problem, the subjects used any cue that seemed relevant.

— When subjects used proof by incomplete enumeration, they demonstrated that they possessed multiple cues. For example, some subjects initially used the causality cue to interpret if-then as AND. However, once they started to evaluate the false antecedent, the nature of the false antecedent triggered the logic cue.

5.1.3. Cueing of If-And-Only-If. The subjects struggled with if-and-only-if more than with any other concept. All but a few subjects failed to even recognize that if-and-only-if was a different construction than if-then. These results indicate that most subjects seemed to lack any cue for the if-and-only-if statement. Consequently, the subjects cued their knowledge based on whether the word "if" or the word "and" grabbed their attention. Subjects who cued their knowledge on "if," reduced to if-then, and subjects who cued their knowledge on "and," reduced to AND. Those subjects who did possess a logic cue for if-and-only-if tended to interpret the statement correctly across all contexts.

5.1.4. Cueing of Complemented Variables. There are two cues for the use of complemented variables: the physical or visual cue and the logic cue (note that the physical cue and visual cue for complemented variables are different from the cues for if-then). The physical cue for complemented variables causes the student to treat Boolean variables as physical analogues of objects. The logic cue caused the student to treat Boolean variables appropriately.

When subjects were asked to interpret specifications for sandwiches or spices, they could easily use a physical cue and think about which items will appear on the sandwich or will be placed in the pie. The subjects thought about which variables should

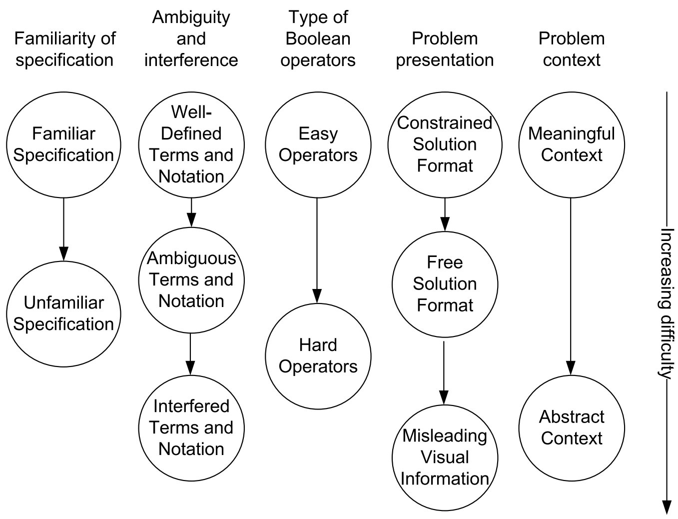


Fig. 11. Model for estimating the difficulty of a Boolean translation task.

appear or be placed in the expression and not about what variables should not appear. For example, if the subjects was thinking visually, "cheese by itself" is the variable c with nothing else connected to it (e.g., $f(c, h, r, t) = c$) rather the correct interpretation $f(c, h, r, t) = c\bar{h}\bar{r}\bar{t}$. Not surprisingly, when subjects were presented with a visual representation of all possible combinations of ingredients, they performed flawlessly: The visual presentation of the problem matched the cue that the subject used.

The presence of a truth table served as a logic cue. A truth table specifically cued the subjects to retrieve their knowledge that absent ingredients must also be included in the Boolean expression. Once subjects were cued to retrieve this knowledge, they used it.

5.1.5. Cueing of Unfamiliar Tasks "Without" and "By-Itself". Subjects used three cues to interpret "without" and "by-itself:" the physical cue of complemented variables (see previous section), an improper (pattern matching) logic cue, and a proper (parsing) logic cue. The improper logic cue was based on proof by incomplete enumeration techniques (in this case, only if-then matched the "without" and "by-itself" expressions). Finally, other subjects cued their knowledge of how to logically parse a sentence.

5.2. A Model for Boolean Translation Problems

Because the goal of grounded theory is to develop a theory that explains a population's behavior, we propose the following model in Figure 11 to summarize what problem features are more likely to cause a student to miscue their knowledge. This model provides a summary of our findings, and it provides a means for estimating the difficulty of interpreting or translating an English specification. The difficulty of a translation task can be thought of as the likelihood that a student will use an improper cue to solve a problem.

In the model, every Boolean translation problem has five attributes: *familiarity of specification*, *ambiguity and interference*, *type of Boolean operator*, *problem presentation*, and *problem context*. Each attribute has a baseline characteristic that minimizes the difficulty of the problem. The difficulty of the problem increases as it incorporates more characteristics farther down the model. The details of this model are preliminary and will be the subject of future research.

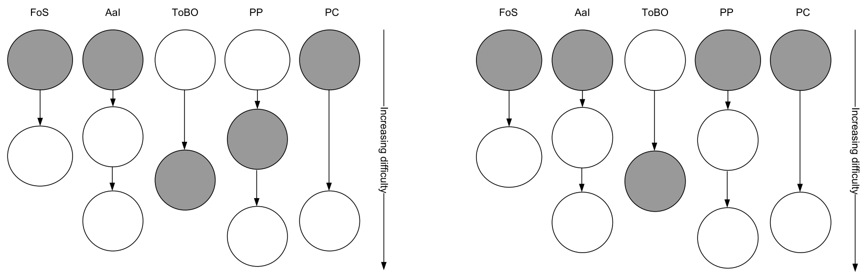


Fig. 12. Difficulty mappings for Boolean translation tasks, "Find a Boolean expression for John's preference: John does not want both apples and bananas (left)." and "Complete the truth table that corresponds to John's preference: John does not want both apples and bananas (right)."

Consider two similar problems: (1) "Find a Boolean expression for John's preference: John does not want both apples and bananas," and (2) "Complete the truth table that corresponds to John's preference: John does not want both apples and bananas." These questions are identical in four characteristics of the model—familiar specification, well-defined terms, hard operators, and meaningful context—and differ on one characteristic—free solution format versus constrained solution format respectively. The model predicts that students would have a higher failure rate with problem (1) than with problem (2). This prediction matches our observations during the interviews. This model is discussed further in a dissertation that encompasses this study [Herman 2011].

5.3. Limitations

Our results may not be generalizable, nor are they intended to be generalized, to all computer science students because all interviewed students were traditional age engineering students from a single institution. Because we found similar misconceptions for students who had taken digital logic classes in two different departments, however, we believe that the results have a degree of generalizability. Future studies should include students from other institutions to test the generality of these findings. The generality of these misconceptions can also be tested through the use of the Digital Logic Concept Inventory (DLCI). Early administrations of the DLCI at multiple institutions have revealed that students across the United States also possess our documented misconceptions. These students revealed their misconceptions at similar rates as the students at the institution where the interviews were conducted [Herman et al. 2011b]. Another limitation of this research was that many of the students (especially the international students) were inarticulate and vague when answering questions. More themes and misconceptions might have been found, but sometimes the student's poor command of spoken English obscured the student's reasoning.

6. CONCLUSION

These results contribute to our understanding of the "what" and "why" of students' difficulties in digital logic. We have documented new conceptual and problem solving difficulties with which students struggle during Boolean translation tasks. We have shown how students fail to understand the purpose of different Boolean logic tools, reduce concepts and tasks, struggle with the ambiguity of the conditional statement, and treat complemented variables as non-essential. We have also explained why students develop these difficulties because of the myriad of context cues. We have shown that students access their knowledge of Boolean logic differently depending on what contextual cues are explicitly present in a problem, but that students manifest correct conceptual understanding more often when the problems help constrain how they cue their knowledge.

The results of this study demonstrate that students who passed digital logic classes with grades of B and C struggle to solve basic conceptual problems even shortly after completing the class. We have modeled these struggles with a five-attribute model that signals what problem attributes are likely to cause miscues. This model can be used to inform the development of standard assessments like our concept inventory, inform future research, and guide instruction.

Our results have confirmed previous findings in the literature, such as students' misconceptions about if-then, and we have supplemented these findings with evidence and explanations for why students develop their misconceptions. We have also identified new misconceptions about other operators that were not previously documented in the literature such as NAND-to-XOR reduction and programming-based misconceptions of if-then.

These misconceptions have been incorporated into the Digital Logic Concept Inventory (DLCI). Administrations of the DLCI have indicated that these misconceptions are common to students from a variety of institutions [Herman et al. 2011b]. Furthermore, results from the DLCI have confirmed that specific instructional paradigms (such as teaching XOR as "one or the other, but not both") may produce unexpected, undesirable side effects on students' understanding of logic concepts [Herman and Handzik 2010].

By testing how students solve similar problems with different contexts and different presentation styles, we demonstrated that students express misconceptions inconsistently, but that they do often possess the correct conception in addition to their misconceptions. Consequently, students' knowledge and use of Boolean logic conceptions is more dependent on the cues that the problem presents rather than the conceptions that they possess. Perhaps most important, students demonstrate that they possess the correct conceptions when the problem format fosters the use of the correct conception.

6.1. Implications for Instruction

Based on these results, we strongly encourage instructors to focus on helping students to identify how they are cueing their knowledge and to practice finding the correct cue to use [Anderson et al. 1998]. For example, instructors might develop exercises where the students are instructed to simply parse the specification, but not translate the specification. If a problem statement contains an "if-and-only-if" statement, students need to simply practice the identification of this important specification. Students easily fail to identify that if-and-only-if has a specific definition in logic tasks. As another example, instructors could require that students practice choosing which translation tool to use. For example, when given a specification to translate, students must pick a tool (e.g., a truth table or Karnaugh map) that they must use before they write the Boolean expression. By forcing students to think about what logic tool to use, the students will be more likely to cue their knowledge according to logic cues rather than cues from colloquial English. In addition, this practice could also help students to choose the appropriate tool for each task (e.g., cue the use of Karnaugh maps for minimization problems). For a final example, instructors might instruct students to identify the cues that they might use when solving a problem, or the instructor can require students to correct mistakes after grading by asking them to identify what cue led them astray.

Instructors can also help students develop appropriate cues (1) by pointing out the different cues that the students might use [Carey 1999] and (2) by providing anchoring examples that help students bridge from colloquial cues to logic cues. For example, when instructors point out the difference between XOR and OR, they often are implicitly pointing out the different cues that the students might use. Because logic students have shown so much facility with the XOR concept, it is clear that this cue training helped the students learn this material. After instructors point out the different cues, they need to provide students with compelling, memorable examples that the students can use as a reference [Clement 1993]. For example, the bar and drinking ages context (Figure 10) bridges the colloquial and logic cues. This bridging example has emotional and logical ties that may help students remember to use this cue when translating English specifications into Boolean specifications.

This training for proper cueing must be coupled with consistent, rigorous modeling of exhaustive proof techniques. Because instructors develop so much facility with translation tasks, they can quickly and accurately choose the correct cue and correctly solve problems. However, this cueing can easily be done tacitly and students will never see what cues their instructors use. Perhaps more dangerous is students' propensity to mimic (however poorly) the behavior of their instructors. When students see their instructors take short cuts and avoid exhaustive enumeration techniques, they also take short cuts to avoid these techniques [Epp 2003]. Instructors must realize that they must foster students' emotional willingness to use appropriate techniques as well as their intellectual assent to use these techniques.

Instructors can foster this emotional willingness by constraining students to use exhaustive enumeration techniques in class, on homework, and on exams [Dufresne et al. 1992]. By requiring students to use situations that prompt the use of correct cues, students are more likely to use these same cues when the bracing structure is removed [Leonard et al. 1996]. These constraints can either be standard abstract representations such as truth tables or more concrete representations such as the pictures of sandwiches. Since both abstract and concrete methods of exhaustive enumeration helped students solve translation tasks correctly, we strongly encourage instructors to emphasize the use of either method in instruction. Future research should investigate whether the abstract or concrete representations (or a combination of the two) of exhaustive enumeration will help students develop the proper cues for solving Boolean translation tasks [Reisslein et al. 2010].

Future research studies with the DLCI, may show which of these instructional suggestions helps students the most. It may also be beneficial to assess students' conceptual knowledge of Boolean logic with the Propositional Logic Test.

In addition to these instructional recommendations, we believe that emphasis on proper logical thinking and complete enumeration of cases will help students in other computer science learning goals such as learning to debug programs and circuits. If students struggle to properly check that all cases satisfy the English specification they were given in logic contexts, how can we expect them to think logically through what test cases are relevant to debugging a program? The ability to translate English specifications into Boolean expressions with rigorous, systematic methods will provide them with valuable analytical thinking skills that can empower students for future learning in computer science and engineering.

ACKNOWLEDGMENTS

Thanks to Joe Handzik and Tabitha Kim for transcribing the interviews.

REFERENCES

ALMSTRUM, V. 1999. The Propositional Logic Test as a diagnostic tool for misconceptions about logical operations. *J. Comput. Math. Sci. Teach. 18*, 3, 205–224.

ALMSTRUM, V. L. 1993. Limitations in the understanding of mathematical logic by novice computer science students. Ph.D. thesis, University of Texas at Austin.

ALMSTRUM, V. L. 1996. Investigating student difficulties with mathematical logic. In *Teaching and Learning Formal Methods*, C. N. Dean and M. G. Hinchey Eds., Academic Press, 131–160.

ALMSTRUM, V. L., HENDERSON, P. B., HARVEY, V., HEEREN, C., MARION, W., RIEDESEL, C., SOH, L.-K., AND TEW, A. E. 2006. Concept inventories in computer science for the topic discrete mathematics. In *Proceedings of the 11th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE'06)*. 163–167.

ANDERSON, J. R., REDER, L. M., AND SIMON, H. A. 1998. Radical constructivism and cognitive psychology. In *Brookings Papers on Education Policy*, D. Ravitch Ed., Brookings Institution, 227–255. (A more complete version, is available online http://act.psy.cmu.edu/ACT/papers/misapplied-abs-ja.html.)

BEN-ARI, M. 2005. The Concorde doesn't fly anymore. In *Proceedings of the Technical Symposium on Computer Science Education (SIGCSE'05)*.

BRANSFORD, J. D., BROWN, A. L., AND COCKING, R. R. 1999. *How People Learn: Brain, Mind, Experience, and School*. National Academy Press, Washington, D.C.

CAREY, S. 1999. Sources of conceptual change. In *Conceptual Development: Piaget's Legacy*, E. K. Scholnick, K. Nelson, S. Gelman, and P. Miller Eds., Erlbaum, Manwah, NJ, 293–326.

CHENG, P. W. AND HOLYOAK, K. J. 1985. Pragmatic reasoning schemas. *Cog. Psych. 17*, 391–416.

CHENG, P. W., HOLYOAK, K. J., NISBETT, R., AND OLIVER, L. 1986. Pragmatic vs. syntactic approaches to training deductive reasoning. *Cog. Psych. 18*, 293–328.

CHI, M. T. H., FELTOVICH, P. J., AND GLASER, R. 1981. Categorization and representation of physics problems by experts and novices. *Cog. Sci. 5*, 121–152.

CLEMENT, J. 1982. Algebra word problem solutions: Thought processes underlying a common misconception. *J. Res. Math. Ed. 33*, 1, 16–30.

CLEMENT, J. 1993. Using bridging analogies and anchoring intuitions to deal with students' preconceptions in physics. *J. Res. Sci. Teach. 30*, 10, 1241–1257.

CLEMENT, J. 2004. A call for action (research): Applying science education research to computer science instruction. *Comp. Sci. Ed. 14*, 4, 343–364.

CLEMENT, J., LOCHHEAD, J., AND SOLOWAY, E. 1980. Positive effects of computer programming on student understanding of variables. In *Proceedings of the ACM Annual Conference (ACM'80)*. 467–474.

DEER, G. W. 1969. The effects of teaching an explicit unit in logic on students' ability to prove theorems in geometry. Ph.D. thesis, Florida State University.

DISESSA, A. A., GILLESPIE, N. M., AND ESTERLY, J. B. 2004. Coherence vs. fragmentation in the development of the concept of force. *Cog. Sci. 28*, 6, 843–900.

DUFRESNE, R., GERACE, W., HARDIMAN, P., AND MESTRE, J. 1992. Constraining novices to perform expert-like problem analyses: Effects on schema acquisition. *J. Learn. Sci. 2*, 307–311.

EPP, S. S. 2003. The role of logic in teaching proof. *Amer. Math. Month. 110*, 10, 886–899.

ERICSSON, K. A. AND SIMON, H. A. 1984. *Protocol Analysis: Verbal Reports as Data.* MIT Press, Cambridge, MA.

EVANS, D. L., GRAY, G. L., KRAUSE, S., MARTIN, J., MIDKIFF, C., NOTAROS, B. M., PAVELICH, M., ROANCOUR, D., REED-RHOADS, T., STEIF, P., STREVELER, R., AND WAGE, K. 2003. Progress on concept inventory assessment tools. In *Proceedings of the 33rd ASEE/IEEE Frontiers in Education Conference (FIC'03).*

GLASER, B. AND STRAUSS, A. 1967. *The Discovery of Grounded Theory: Strategies for Qualitative Research.* Aldine, Chicago, IL.

GOETTING, M. M. 1995. The college student's understanding of mathematical proof. Ph.D. thesis, University of Maryland.

GOLDMAN, K., GROSS, P., HEEREN, C., HERMAN, G., KACZMARCZYK, L., LOUI, M. C., AND ZILLES, C. 2010. Setting the scope of concept inventories for introductory computing subjects. *ACM Trans. Comput. Ed. 10*, 2, 1–29.

GREENE, S. L., DEVLIN, S. J., CANNATA, P. E., AND GOMEZ, L. M. 1990. No ifs, ands, or ors: A study of database querying. *Int. J. Man-Mach. Studies 32*, 303–326.

HAKE, R. 1998. Interactive-engagement vs. traditional methods: A six-thousand-student survey of mechanics test data for introductory physics courses. *Amer. J. Phys. 66*, 64–74.

HARDIMAN, P. T., DUFRESNE, R. J., AND MESTRE, J. P. 1989. The relation between problem categorization and problem solving among novices and experts. *Mem. Cogn. 17*, 627–638.

HAREL, G. AND SOWDER, L. 1998. Students' proof scheme. In *Research in Collegiate Mathtematics Education*, Vol. III, E. Dubinsky, A. Schoenfeld, and J. Kaput Eds., American Mathematical Society, 234–283.

HERMAN, G. L. 2011. The development of a digital logic concept inventory. Ph.D. thesis, University of Illinois at Urbana-Champaign.

HERMAN, G. L. AND HANDZIK, J. 2010. A preliminary pedagogical comparison study using the Digital Logic Concept Inventory. In *Proceedings of the 40th ASEE/IEEE Frontiers in Education Conference (FIC'10)*.

HERMAN, G. L., KACZMARCZYK, L., LOUI, M. C., AND ZILLES, C. 2008. Proof by incomplete enumeration and other logical misconceptions. In *Proceedings of the 4th International Workshop on Computing Education Research (ICER'08)*. 59–70.

HERMAN, G. L., ZILLES, C., AND LOUI, M. C. 2009. Work in progress: Students' misconceptions about state in digital systems. In *Proceedings of the 39th ASEE/IEEE Frontiers in Education Conference (FIC'09)*.

HERMAN, G. L., LOUI, M. C., AND ZILLES, C. 2010. Creating the Digital Logic Concept Inventory. In *Proceedings of the 41st Annual ACM Technical Symposium on Computer Science Education (ITiCSE'10)*. 102–106.

HERMAN, G. L., LOUI, M. C., AND ZILLES, C. 2012. Flip-flops in students' conceptions of state. *IEEE Trans. Ed.*

HERMAN, G. L., LOUI, M. C., AND ZILLES, C. 2011a. Students' misconceptions about medium-scale integrated circuits. *IEEE Trans. Ed. 54*, 4, 637–645.

HERMAN, G. L., ZILLES, C., AND LOUI, M. C. 2011b. Administering a digital logic concept inventory at multiple institutions. In *Proceedings of the American Society for Engineering Education Annual Conference and Exposition (ASEE'11)*.

HERMAN, G. L., ZILLES, C., AND LOUI, M. C. 2011c. How do students misunderstand number representations? *Comput. Sci. Ed. 23*, 3, 289–312.

HESTENES, D. AND HALLOUN, I. A. 1995. Interpreting the Force Concept Inventory: A response to March 1995 critique by Huffman and Heller. *Phys. Teach. 33*, 502–506.

HESTENES, D., WELLS, M., AND SWACKHAMER, G. 1992. Force Concept Inventory. *Phys. Teach. 30*, 141–158.

HOC, J. M. 1989. Do we really have conditional statements in our brain. In *Studying the Novice Programmer*, E. Soloway and J. C. Spohrer Eds., Lawrence Erlbaum Associates, Hillsdale, NJ, 179–190.

JUNGWIRTH, E. 1985. Science teaching and pupil avoidance of logical fallacies. *South African J. Ed. 5*, 55–60.

JUNGWIRTH, E. 1987. Avoidance of logical fallacies - A neglected aspect of science-education and science teacher education. *Res. Sci. Technol. Ed. 5*, 43–58.

JUNGWIRTH, E. 1990. Science teachers spontaneous, latent, or nonattendance to the validity of conclusions in reported situations. *Res. Sci. Technol. Ed. 8*, 103–115.

KIM, Y. 1995. The reasoning ability and achievement of college level students enrolled in a logic class in computer science. Ph.D. thesis, University of Texas at Austin. http://www.lib.utexas.edu/indexes/titles.php?id=114.

KVALE, S. 1996. *Interviews: An Introduction to Qualitative Research Inquiry.* Sage, Thousand Oaks, CA.

LEONARD, W. J., DUFRESNE, R. J., AND MESTRE, J. P. 1996. Using qualitative problem-solving strategies to highlight the role of conceptual knowledge in solving problems. *Amer. J. Phys. 64*, 1495–1503.

LONGINO, J. T., LOUI, M. C., AND ZILLES, C. 2006. Student misconceptions in an introductory digital logic design course. In *Proceedings of the American Society for Engineering Education Annual Conference and Exposition (ASEE'06)*.

MCCRACKEN, M., ALMSTRUM, V., DIAZ, D., GUZDIAL, M., HAGAN, D., KOLIKANT, Y. B.-D., LAXER, C., THOMAS, L., UTTING, I., AND WILUSZ, T. 2001. A multi-national, multi-institutional study of assessment of programming skills of first-year CS students. In *Proceedings of the 11th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education Working Group Reports (ITiCSE'06)*. 125–180.

MERRIAM, S. B. 2009. *Qualitative Research: A Guide to Design and Implementation.* Jossey-Bass.

MILES, M. B. AND HUBERMAN, M. 1984. *Qualitative Data Analysis: A Sourcebook of New Methods.* Sage Publications, Thousand Oaks, CA.

NISBETT, R. E. 1993. *Rules for Reasoning.* Lawrence Erlbaum Associates, 1–12.

OWENS, B. B. AND SEILER, J. 1996. A study of the relationship between performance in propositional logic and computer science. *J. Comput. Small Coll. 11*, 7, 90–93.

PANE, J. AND MYERS, B. 2000. Tabular and textual methods for selecting objects from a group in Institute for Software Research.

PERKINS, D. 1999. The many faces of constructivism. *Ed. Lead. 57*, 3, 6–11.

PERKINS, D. N. AND MARTIN, F. 1986. Fragile knowledge and neglected strategies in novice programmers. In *Proceedings of the 1st Workshop on Empirical Studies of Programmers*. E. Soloway and S. Iyengar Eds., Norwood, NJ, 213–229.

PETRUSHKA, D. 1984. A study of the effect of content on the ability to do syllogistic reasoning: An investigation of transferability and the effect of practice. Ph.D. thesis, Rutgers University.

PIBURN, M. D. 1989. Reliability and validity of the Propositional Logic Test. *Ed. Psych. Measure. 49*, 667–672.

PIBURN, M. D. 1990. Reasoning about logical propositions and success in science. *J. Res. Sci. Ed. 27*, 9, 887–900.

REISSLEIN, M., MORENO, R., AND OZOGUL, G. 2010. Pre-college electrical and computer engineering instruction: The impact of abstract vs. contextualized representation and practice on learning. *J. Eng. Ed. 99*, 3, 225–235.

SCHOENFELD, A. H. 1992. Learning to think mathematically: Problem solving, metacognition, and sense-making in mathematics. In *Handbook for Research on Mathematics Teaching and Learning.* MacMillan, New York, 334–370.

SELDEN, J. AND SELDEN, A. 1995. Unpacking the logic of mathematical statements. *Ed. Studies Math. 29,* 2, 123–151.

STAGER-SNOW, D. 1985. Analytical ability, logical reasoning, and attitude as predictors of success in an introductory course in computer science for non-computer science majors. Ph.D. thesis, Rutgers University.

STRAUSS, A. AND CORBIN, J. 1998. *Basics of Qualitative Research*. Sage, Thousand Oaks, CA.

TEW, A. E. AND GUZDIAL, M. 2010. Developing a validated assessment of fundamental CS1 concepts. In *Proceedings of the 41st Annual ACM Technical Symposium on Computer Science Education (CSE'10)*. 97–101.

THE COMPUTER SOCIETY OF THE INSTITUTE FOR ELECTRICAL AND ELECTRONIC ENGINEERS AND ASSOCIATION FOR COMPUTING MACHINERY. 2001. Computing Curricula 2001, Computer Science Volume. http://www.sigcse.org/cc2001/index.html.

VANDEGRIFT, T., BOUVIER, D., CHEN, T.-Y., LEWANDOWSKI, G., MCCARTNEY, R., AND SIMON, B. 2010. Commonsense computing (episode 6): Logic is harder than pie. In *Proceedings of the 10th Koli Calling International Conference on Computing Education Research (KOLI'10)*. 76–85.

VERSCHAFFEL, L., DECORTE, E., AND LASURE, S. 1994. Realistic considerations in mathematical modeling of school arithmetic word problems. *Learn. Instruc. 4,* 273–295.

VOSNIADOU, S. AND BREWER, W. F. 1992. Mental models of the earth: A study of conceptual change in childhood. *Cog. Psych. 24*, 4, 535–585.

WASON, P. C. 1966. *New Horizons in Psychology I*. Penguin, Harmondsworth, UK.

WEISS, R. S. 1994. *Learning from Strangers: The Art and Method of Qualitative Interview Studies*. The Free Press, New York, NY.

WOLLMAN, W. 1983. Determining the sources of error in a translation from sentence to equation. *J. Res. Math. Ed. 14*, 3, 169–181.

Received November 2010; revised April 2011, July 2011, August 2011; accepted August 2011