

*PROPOSAL FOR FSSR STATUS FOR A REVISED FSSR COURSE*  
(September 27, 2005)

Part A: Cover Application

NOTE: The revision of CMSC 101 Minds and Machines described here increases the number of credit hours from 3 to 4. (A separate proposal to increase the credit hours was approved by the Department of Mathematics and Computer Science and then sent to Dean Newcomb's office on September 13, so that his office can send it on to academic council.) The existing 3-credit course was approved approximately 9 years ago as a Symbolic Reasoning (FSSR) course and has been taught several times since then. The revised course *increases* the expectations concerning the overall strength of the course as it relates to the FSSR requirement, as explained in Part B.

- *Proposed field of study:* Symbolic Reasoning
- *Course number:* Computer Science 101
- *Course title:* Minds and Machines
- *Catalog description:* Formal deduction in propositional logic. The fundamentals of computer architecture. An elementary exploration of the extent to which symbolic reasoning can be automated, including a consideration of related results in fields such as neuroscience and artificial intelligence. No prerequisites. 4 hours credit.
- *Course prerequisite(s):* none
- *Number of credit hours:* 4
- *Typical estimated enrollment:* 22 students per section. (This upper limit on enrollment is necessary, since the course is offered in one of the computerized teaching labs within Jepson Hall that has 24 workstations. It is not unusual for 2 of those workstations to fail to work during a typical class session.)
- *How often and by whom will the course be offered?* Arthur Charlesworth, or another member of the computer science faculty, will teach the course. Charlesworth has taught the existing 3-hour CMSC 101 course during four previous semesters. One or two sections of the course will be offered each year.
- *Staffing implications for the school/department/unit:* Will have no effect on number of staff. This course simply replaces the current CMSC 101 course. Fortunately, the current size of the computer science faculty, when compared with the current demands by computer science majors for computer science major courses, enables us to offer such a non-major course

more frequently than in the past. This enables students from a variety of majors to gain insights into questions related to the nature of minds and machines.

- *Adequacy of library, technology, and other resources.* All of the University's resources relevant to this course are adequate.
- *Any interdepartmental and interschool implications:* This revised course has the same relationship to other courses, curricula, etc., as the current CMSC 101 course.
- *Contact person:* Arthur Charlesworth.

PROPOSAL FOR FSSR STATUS FOR A REVISED FSSR COURSE  
(September 27, 2005)

Part B: Separate Detailed Explanation

- *How the course fulfills the purpose of the Symbolic Reasoning field of study.*  
The revision of CMSC 101 Minds and Machines described here increases the number of credit hours from 3 to 4. The existing 3-credit course was approved approximately 9 years ago as a Symbolic Reasoning (FSSR) course and has been taught several times since then. The revised course *increases* the expectations concerning the overall strength of the course as it relates to the FSSR requirement, as follows. The revised course continues to cover the key symbolic reasoning concepts of the 3-credit course: computer software’s symbolic nature, and a concentrated focus on formal deduction in propositional logic. In addition, the revised course increases the course expectations to ensure that students have an opportunity to better understand the impact and limitations of *all forms* of rigorous symbolic reasoning. The greater expectation for the course involves the proof of the unsolvability of the halting problem, the coverage of Gödel’s Incompleteness Theorem, and a consideration of concepts of neuroscience and artificial intelligence related to symbolic reasoning and the nature of minds and machines.

Here is how the key aspects of the 3-credit FSSR CMSC 101 course related to Symbolic Reasoning continue in the revised 4-credit course. The course provides an elementary exploration of the question: “to what extent can symbolic reasoning be automated?” In addition to addressing this question, two main topics provide background for addressing the question. These topics are:

- *What is “automation”?* Students obtain a concrete answer by learning how English language problems (such as the logical decisions made by a simple timing-clock and by a cruise control) are translated into computer programs, how those computer programs are translated by an algorithm (that the students learn) into programs that can actually directly execute on a computer, and how the resulting “machine language” programs are executed by an algorithm (that the students learn).
- *What is “symbolic reasoning”?* Students obtain a concrete answer through their own experience with formal deduction in propositional logic. Students translate suitable English language arguments into the symbolic system of propositional logic. Students use formal inference rules to derive the conclusion from the premises, when this is logically possible. The goal is for students to be able to create both translations and proofs such as the one illustrated on the next page,

Real-world problem	Real-world solution
Assuming only the following: 1) If government deficits continue at their present rate and a recession sets in, then interest on the national debt will become unbearable and the government will default on its loans. 2) If a recession sets in, then the government will not default on its loans. Can one conclude the following? Either government deficits will not continue at their present rate or a recession will not set in, or both.	Yes

translation of  $\vee$  problem

interpretation of  $\wedge$  solution

Logic problem	Logic solution
1. (C.R) $\rightarrow$ (I.D) 2. R $\rightarrow$ $\sim$ D To show: $\sim$ C $\vee$ $\sim$ R	1. (C.R) $\rightarrow$ (I.D) 2. R $\rightarrow$ $\sim$ D To show: $\sim$ C $\vee$ $\sim$ R  Justification <hr/> 3.   C ACP 4.     R AIP 5.     C.R 3, 4 Conj 6.     I.D 1, 5 MP 7.     D.I 6 Com 8.     D 7 Simp 9.     $\sim$ D 2, 4 MP 10.     D. $\sim$ D 8, 9 Conj 11.   $\sim$ R 4-10 IP 12. C $\rightarrow$ $\sim$ R 3-11 CP 13. $\sim$ C $\vee$ $\sim$ R 12 Impl

where a period denotes “and”,  $\vee$  denotes “or”, etc., ACP justifies an assumption to support a conditional proof, AIP justifies an assumption to support an indirect proof, CP is justification by conditional proof, IP is justification by indirect proof, and the other justifications are formal rules of inference.

The course thus provides a framework, not only for studying the nature of symbolic reasoning, but for experiencing it in depth.

Since the existing 3-hour version of this course was developed by the submitter of this proposal, and since that same person helped to craft what is now the catalog description of the Symbolic Reasoning field of study requirement, it is straightforward to explain how the revised 4-hour version of the course satisfies each part of that description (which appears below in italics for ease in reading).

*This field of study emphasizes symbolic problem solving, a process that includes translating problems into terms that are amenable to treatment within a symbolic system;*

The translation of problems into computer programs, such as for a timing-clock and a cruise control, satisfies that requirement. The symbolic logic translation process illustrated on the preceding page, which forms the core of student learning in the course (and which continues to constitute a major part of student work in the 4-hour course), satisfies that requirement.

*understanding consistent rules by which the information relevant to the problem may be processed in order to obtain a solution;*

The rules for translating computer programs into programs that can actually execute on a computer are consistent rules, as are the rules for actually executing the resulting machine language programs. The formal “inference rules” used in logic for justifying each step in a proof, illustrated on the preceding page, are consistent rules.

*recognizing important underlying principles that govern the application of these rules;*

Students are not only taught those underlying principles but are taught them in a way that helps them understand why the rules “make sense”. For computer programs, this is accomplished by reminding students of the overall process (from the intuitive problem to solve through each step to the execution of the machine code) and by pointing out how the later parts of that work relate directly (but in a much lower level form) to earlier parts of that work. For logical proofs, this is accomplished by first covering logical concepts in an intuitive way and then, during the weeks of work with symbolic

logic, reminding students how that later work relates directly (but in a much more rigorous form) to their earlier intuitive understanding.

*and judging both the appropriateness of known solution methods to a particular problem and the quality of reasonableness of the solution obtained.*

Students learn that information must be discretized in order for one to use computer programs to process that information, and students learn of the variety of kinds of information that can be discretized (more technically speaking, can be coded as bits), yet students become aware that it may not always be possible to discretize all important kinds of information (such as, say, consciousness or emotions). Also in the context of computer programs, students learn not to assume that the output of a computer program is necessarily reasonable, since humans often make errors when writing computer programs and the most trivial error in a computer program often results in that program producing outputs that are significantly in error. The higher expectation of the 4-hour course goes much deeper into the limitation of computer programs, by expecting students to understand the proof of the unsolvability of the halting problem. That proof is beyond the scope of most (perhaps all) introductory computer science courses, and will be covered by a separate handout written specifically for this course, tailored to use rules of inference the students have studied.<sup>1</sup> For logical proofs, the student's intuitive understanding of logic (described above) helps make clear the reasonableness of applying the formal rules of inference to propositional logic. The higher expectation of the 4-hour course goes much deeper into the nature of and limitations of any rigorous form of logic, through the coverage of Gödel's Incompleteness Theorem, which is beyond the scope of most (perhaps all) textbooks for teaching introductory logic, and thus requires a separate paper written specifically for this course.

The second paragraph of the catalog description of Symbolic Reasoning emphasizes that the use of software tools should not be the primary objective of the course. In this course (both the 3-hour and the enhanced 4-hour version), no automated software tools assist the actual symbolic-reasoning problem-solving by students. (Students do

---

<sup>1</sup>This course is directly related to a research area of the developer and proposer of both the current 3-hour course and this 4-credit course. Charlesworth has published two articles in journals of the Mathematical Association of America on proving the unsolvability of the halting problem and on proving Gödel's Incompleteness Theorem. He has also published, in the last three years, two additional articles in journals of the Association for Computing Machinery. The new articles extend those two proofs, and the latest article extends this research project into a technical examination of the nature of automated symbolic reasoning systems.

use a software system to *check* their solutions to formal logic proofs. That software system gives them helpful, immediate feedback on their success or lack thereof, but the solutions themselves are obtained by the students and not by that software system.)

In addition to fully satisfying the catalog description of Symbolic Reasoning, the 4-hour version of this course goes significantly beyond that description. The parts of the enhanced course related to the halting problem, Gödel's Theorem, neuroscience, artificial intelligence, and related topics, enable students to see unusually deep (for a 100-level course) into the nature of symbolic reasoning. The enhanced course raises important questions about the role that symbolic reasoning plays within a much broader context.

- *Course objectives:* The objective of the course is to explore questions related to minds and machines, while developing student ability to carry out symbolic reasoning and critical thinking. The questions explored include the following, which are stated here perhaps as simply as possible:
  - What are the logical principles behind how computers work?
  - Are there inherent limits to the reasoning of computers, even within the realm of computational problems?
  - What is the nature of fully rigorous symbolic reasoning?
  - Are there purely logical, inherent limits to rigorous symbolic reasoning?
  - What is the nature of current attempts to achieve artificial intelligence?
  - Can one define the concept of thinking so it doesn't presuppose a biological origin?
  - Will computers ever be able to think?
  - What, if anything, are the essential differences between minds and computers?
  - Will it ever be possible for humans to fully understand the human brain as operating in a computational fashion?

Such questions are among central questions appropriate for consideration within the liberal arts. For some of these questions, students will learn and discover possibly surprising answers. For other questions the answers are not yet known, and conscientious students will leave this introductory course better able to address such questions, giving them a foundation for life-long insights about the relationship between humans and computers.

• *Proposed syllabus, including reading list*

1. Introduction to computer architecture and language translation  
(Textbook: *How Computers Work*, by Charlesworth, Chapters 1 through 7 and Chapter 10.)
  - The syntax and semantics of a simple but fully powerful symbolic instruction set
  - An algorithm for translating any program (written using the given instruction set) into machine code programs
  - An algorithm for executing machine code programs
  - A broader view  
Comparison of the machine studied with other computers, of the instruction set studied with other programming languages, and of the translation process with other automatic translation processes from computer programs to machine code.
  - Proof of a fundamental limitation to computational machines  
(Source: “The Unsolvability of the Halting Problem”, by Charlesworth about Turing’s theorem)
2. Formal symbolic reasoning  
(Textbook: *A Concise Introduction to Logic*, by Hurley; emphasis on Chapters 1, 6, and 7.)
  - Basic concepts of logic
  - Devising truth tables in propositional logic
  - Creating formal proofs in propositional logic  
This is the central part of the course, from the standpoint of developing student ability in symbolic reasoning. Students learn to discover their own formal proofs through a series of numerous and progressively more challenging homework assignments. The full eighteen rules of inference (such as De Morgan’s Rule and Modus Ponens) are covered as well as the method of conditional proof and the method of indirect proof.
  - A broader and deeper view: predicate logic and beyond  
(Source: “An Introduction to Gödel’s Incompleteness Theorem”, paper by Charlesworth.)
3. Brief introduction to the nature of artificial intelligence  
(Sources include course handouts and information from articles by Turing, the Churchlands, Searle, and Hofstadter.) Topics include simple exercises to help students understand the difference between an artificial neural net approach to AI and a symbolic-logic approach to AI, and a consideration of the fact that some reasoning that is difficult for humans has been achieved by AI, yet – despite decades of research efforts – most reasoning that is far easier for humans is significantly beyond the current state of the art in AI.



4. Brief introduction to neuroscience  
(Sources: “A Quick Look at the Human Brain”, paper by Charlesworth; “Can the Mind Just Be a Machine?”, video lecture by Bertil Hille of the University of Washington.) Topics include low-level processing – such as propagated action potentials and synaptic transmission – and the higher-level compartmentalization of the brain into differently functioning components.

5. Other topics as time permits  
The additional topics include a consideration of the apparent contrast between the intelligence of honey bees and that both of the Sphex and of the Australian mud wasp, the simulation of apparently algorithmic wasp reasoning by a program written using the set of programming language instructions studied in the course, simple classroom experiments related to the topics covered in the course, and course-related topics suggested by news accounts occurring during the semester.

- *Details of how the course will be taught:* Student grades will be based largely on a final exam, two major tests, and approximately four quizzes. To help prepare them for such tests and quizzes, students will be expected to work and turn in homework problems at nearly every class session, and most of the homework assignments will focus on developing (over the course of most of the semester) student ability to succeed in a creative (non-algorithmic) way with symbolic reasoning as that relates to the logic component of the course. Some of the additional homework assignments will focus instead on developing student ability to master concepts related to algorithmic thinking as that relates to the automation of symbolic reasoning by computers, so that students experience first-hand the contrast between the *obviously* algorithmic way computers typically process information and the way minds perform symbolic reasoning. Other homework assignments will focus, in other ways, on developing student understanding of the motivating course questions related to minds and machines listed earlier in this proposal, thereby encouraging students to think deeply and broadly about significant issues related to symbolic reasoning. Some homework assignments will involve other topics related to minds and machines, such as neuroscience and artificial intelligence, and at times students will be expected to work two different kinds of homework assignments from one class session to the next.

*Summary of increased expectations, compared with the previously approved 3-credit course:* The depth and breadth of the 4-credit course is greater. The greater depth of the course is due in part to the addition of mathematical topics related to symbolic reasoning, such as the proof of the unsolvability of the halting problem and the nature of Gödel’s Incompleteness Theorem. The greater breadth of the course (and, to some extent, also the greater depth) is due

to the addition of some material from the fields of neuroscience and artificial intelligence. For the latter reason, the revised course replaces the following sentence in the current catalog description:

An elementary exploration of the extent to which symbolic reasoning can be automated.

with the sentence:

An elementary exploration of the extent to which symbolic reasoning can be automated, including a consideration of related results in fields such as neuroscience and artificial intelligence.

This sentence is flexible in order to permit the inclusion in the future of some material from such fields as cognitive science and robotics.