

Intelligent Systems Study Guide

Don Colton
Brigham Young University–Hawaii

April 6, 2015

Preface

This is the official study guide for the CS 490R class, Intelligent Systems, as taught by Don Colton, Winter 2015. It is focused directly on the grading of the course.

<http://byuh.doncolton.com/cs440/2151/sguide.pdf> is the study guide, which is this present document. **It may be updated at any time throughout the semester**, as new assignments are clarified and material is added.

Syllabus

<http://byuh.doncolton.com/cs440/2151/syl.pdf> is the official syllabus for this course. It is largely reproduced in Chapter 2 (page 8) below.

Study Guide

The content of this study guide is centered on three projects and two skills that will be main parts of the curriculum of this course. It also contains typical final exam questions.

The skills are (a) working with conditional probabilities, and (b) solving propositional logic systems by resolution.

The projects are (a) robotic vacuum cleaner, (b) Wumpus hunter, and (c) speech recognition of numbers.

Each skill or project is the subject of a chapter that provides background and training.

Test Bank

As material is covered in the book, exam questions are inserted to show what the student should be learning. These exam questions appear throughout the book, together with answers. At the back of the book, Appendix B (page 89) is a Test Bank. It repeats these same questions that appeared throughout the book, but without their answers.

The Test Bank is a way for students to test themselves by reviewing the questions and making sure they know at least one acceptable answer.

The Test Bank is also a way for teachers to be reminded of specific things that students should be able to answer.

Sometimes the questions and answers summarize material that is presented nearby in greater detail.

Sometimes the questions and answers are the actual presentation of that material. This is especially true when the specific material is something simple like vocabulary, and repetition would be tiresome and redundant.

Following is the format in which questions and answers are presented.

Exam Question 1 (p.89): What does AI stand for?

Acceptable Answer: artificial intelligence

The questions are linked to make it easy for the student to jump back and forth between the test bank and the content chapters.

Contents

1	Introduction	5
2	Syllabus Extracts	7
3	Readings	35
4	Conditional Probability	36
5	Bayesian Probability	43
6	Propositional Resolution	47
7	Projects in General	52
8	Vacuum	55
9	Hunt the Wumpus	60
10	Number Recognition	64
11	Phonetics	72
12	Exam Topics	73
A	CC2001: Intelligent Systems	80

B Test Bank

89

Index

97

Chapter 1

Introduction

Contents

1.1 Sending Me Email	5
--	---

1.1 Sending Me Email

These rules apply to all emails related to this class.

Rule One: Send emails to **doncolton2@gmail.com**

Rule Two: Put **cs490r** in your email subject line.

Rule Three: Single-topic emails get answered faster than multi-topic emails do. Avoid combining several topics in a single email.

I must confess, I sometimes get buried in email. I do not want to overlook your email to me, or have it end up caught in my spam filter. And for my own sanity I want to be able to find and deal with all the email related to this class at the same time. This is especially true for large classes.

My solution is to have you put **cs490r** in your email subject line, preferably as the first word. If you do this, my email system will immediately and automatically respond to you, telling you that I got your email and it is in my queue.

If you fail to do this, you will not get an immediate reply and your email will end up in some other queue in my work flow. Your email will not be noticed when I am grading for this class. In the best case I will read your

email and ask you to send it again with the correct word in the subject line.
In the worst case your email will be in my spam folder and I will never even
see it.

You have been warned.

Chapter 2

Syllabus Extracts

Contents

2.1	Overview	8
2.1.1	Expected Proficiencies	9
2.2	Course and Faculty	9
2.2.1	Course Information	9
2.2.2	Faculty Information	9
2.2.3	Course Readings and Materials	10
2.3	Calendar	11
2.4	Grading	12
2.4.1	CS 490R Grade Books	12
2.4.2	Attendance (26 points)	13
2.4.3	Readings (176 pts)	13
2.4.4	Skills Tests (200 pts)	14
2.4.5	Programming Labs (600 pts)	14
2.4.6	Final Exam Day	15
2.4.7	Extra Credit	15
2.5	Instructional Methods	16
2.5.1	BYUH Learning Framework	16
2.5.2	Prepare for CS 490R	16
2.5.3	Engage in CS 490R	17
2.5.4	Improve at CS 490R	17
2.5.5	Support	17
2.5.6	Office Hour / Open Lab	17

2.5.7	Study Groups	18
2.5.8	Tutoring	18
2.6	Course Policies	18
2.6.1	Special Treatment	19
2.6.2	Reasonable Accommodation	19
2.6.3	Communication	20
2.7	Learning Outcomes	22
2.7.1	ILOs: Institutional Outcomes	22
2.7.2	PLOs: Program Outcomes	23
2.7.3	CLOs: Course Outcomes	26
2.8	General Topics	26
2.8.1	Academic Integrity	26
2.8.2	Sexual Misconduct	29
2.8.3	Dress and Grooming Standards	30
2.8.4	Accommodating Special Needs	31
2.9	Syllabus Summary	32

The syllabus is intended to be stable and reliable. I publish the syllabus near the start of the semester, and then I do not change it in any way, except in case of extreme emergency.

The study guide, on the other hand, gets changed and updated throughout the semester.

This chapter of the study guide simply repeats, for your convenience, and for indexing purposes, the material given in the syllabus. In case it differs from the syllabus in any important way, the syllabus takes precedence. However, the study guide version may have corrections of typographical errors, etc.

2.1 Overview

Artificial Intelligence (AI) has been a goal of computer scientists since the earliest days of computing. It has benefited and suffered from overblown expectations. It has had seasons of high respect and great disdain. Every computer scientist needs a basic understanding of AI, including its terminology and its accomplishments.

2.1.1 Expected Proficiencies

As you begin this course, we assume you can program in at least one language available under Linux. We require that programs in your chosen language can be run from the command line. There will be three programming tasks that require this skill. Commonly students use either Perl, Java, or C++ for these tasks.

We assume that you are familiar with the content usually learned in an Algorithms class, including manipulation of lists, trees, and graphs, and including search (breadth first, depth first).

Some of the programming activities would benefit if you are comfortable with regular expressions.

2.2 Course and Faculty

2.2.1 Course Information

- **Title:** Intelligent Systems
- **Course Number:** CS 490R, formerly CS 440.
- **Course Description:** (from the catalog) Fundamental issues in intelligent systems, search and constraint satisfaction, knowledge representation and reasoning.
- **Prerequisites:** CS 301
- **Semester/Year:** Winter 2015
- **Semester Code:** 2151
- **Meeting Time:** TuTh 09:20 to 10:50
- **Location:** GCB 101
- **First Day of Instruction:** Tue, Jan 13
- **Last Day to Withdraw:** Tue, Mar 10
- **Last Day of Instruction:** Tue, Apr 14
- **Final Exam:** Thu, Apr 16, 10:00–12:50

2.2.2 Faculty Information

- **Instructor:** Don Colton
- **Office Location:** GCB 128
- **Office Hours:** MWF 14:30-15:00, TuTh 15:30-16:00.

-
- **Email:** doncolton2@gmail.com
 - **Campus Homepage:**
<http://byuh.doncolton.com/> is my campus homepage. It has my calendar and links to the homepages for each of my classes.
 - **Off-Campus Homepage:**
<http://doncolton.com/> is my off-campus homepage.

I have reserved GCB 111 on MWF from 14:30 to 15:30 (right after my CIS 101 class) and on TuTh from 15:30 to 17:00 (right after my IT 240 class) so my students (and others) can study in a lab setting and meet with me and each other. I will be there at the start of those hours, and will stay as long as students are asking me questions. I also allow the room as an Open Lab for your use either individually or in groups, for my class or for other classes.

2.2.3 Course Readings and Materials

- **Textbook:** <http://aima.cs.berkeley.edu/>
(Rental) *Artificial Intelligence, A Modern Approach, Second Edition*, by: Stuart Russell and Peter Norvig 1080 pages. ISBN: 0-13-790395-2. Prentice Hall.
- **Learning Management System:**
<https://dcquiz.byuh.edu/> is the learning management system for my courses.
- **Course Homepage:**
<http://byuh.doncolton.com/cs440/> is my course homepage. It has links to many things including the syllabus and study guide.
- **Study Guide:**
<http://byuh.doncolton.com/cs440/2151/sguide.pdf> is the study guide for this course. It is indexed and includes a copy of almost all of this syllabus. The study guide may be updated during the semester as assignments and/or deadlines are updated.

The textbook is a department rental. The third edition has been published but we will continue to use the second edition because it is adequate for our needs. If you wish, you can purchase the second or third edition (through Amazon, for instance) and use it instead of renting the second edition.

The textbook views the AI field as having a goal to build intelligent agents.

I think this framework makes it easier to understand what AI really is and is not.

If you plan to study Artificial Intelligence at graduate school, you are strongly encouraged to dig deeply into the textbook. It is excellent.

2.3 Calendar

R1 through R11 are readings.

Tue Jan 13 Syllabus, CP (Cond Probability)
Thu Jan 15 CP, q45 Exam (10:10 to 10:50)
Tue Jan 20 R1
Thu Jan 22 CP, q45 Exam (10:10 to 10:50)
Tue Jan 27 R2, VA (Vacuum Agent)
Thu Jan 29 VA bake off
Tue Feb 03 R3, VA
Thu Feb 05 VA bake off
Tue Feb 10 R4, VA
Thu Feb 12 VA bake off
Tue Feb 17 R5, Res (Resolution)
Thu Feb 19 Res Exam q41 (10:10 to 10:50)
Tue Feb 24 R6, Res
Thu Feb 26 Res Exam q41 (10:10 to 10:50)
Tue Mar 03 R7, WA (Wumpus Agent)
Thu Mar 05 WA bake off
Tue Mar 10 R8, WA
Tue Mar 10 Last Day To Withdraw
Thu Mar 12 WA bake off
Tue Mar 17 R9, WA
Thu Mar 19 No Class: Empower Your Dreams
Tue Mar 24 WA bake off (no readings)
Thu Mar 26 No Class: Kuhio Day
Tue Mar 31 R10, NR (Numbers Recognition)
Thu Apr 02 NR bake off
Tue Apr 07 R11, NR
Thu Apr 09 NR bake off
Tue Apr 14 NR bake off
Thu Apr 16 Final, 10:00–12:50, GCB 101

We meet 26 times including the final exam. The dates listed are not likely to change.

The Final Exam date is established by the University and is firm. It will not change unless there is a fire or a flood or something.

2.4 Grading

I use a 60/70/80/90 model based on 1000 points.

Based on 1000 points

930+	A	900+	A-	870+	B+
830+	B	800+	B-	770+	C+
730+	C	700+	C-	670+	D+
630+	D	600+	D-	0+	F

The 1000 points are divided up as follows:

- Attendance (26 points)
- Readings and Oral Reports (176 points)
- Exam: Conditional Probability (100 points)
- Exam: Resolution (100 points)
- Lab: Vacuum Agent (200 points)
- Lab: Wumpus Agent (200 points)
- Lab: Numbers Recognition (200 points)

2.4.1 CS 490R Grade Books

In my Learning Management System (DCQuiz), I keep several online grade books so you can see how your points are coming along. This lets you compare yourself with other students in the class (without seeing their names).

2151 CS 490R Overall Grade Book: This includes the totals from all the other grade books. This is where you can find your final grade at the end of the course.

2151 CS 490R (whatever) Grade Book shows your points in the (whatever) category. (whatever) is Attendance or Readings.

2.4.2 Attendance (26 points)

Each day in class starts with the “daily update” (DU). It is my way of reminding you of due dates and deadlines, sharing updates and news, and taking roll. It is your way of saying something anonymously to each other and to me. It must be taken in class at a classroom computer during a window of time that starts a few minutes before class and ends 2 minutes into class.

Attendance Policy: You must attend to complete the Daily Update and thereby earn the Attendance points. You must attend to give oral reports on your readings. You must attend to take the Skills Tests and thereby earn the Skills Test points. Besides that, there is no penalty for being late or lack of attendance.

2151 CS 490R Attendance Grade Book shows your attendance points, one point per day, for 26 days. You get one point for each time you do the daily update. If you arrive too late to complete the daily update, you will not receive the attendance point for that day.

Tardiness: My tardiness policy is that you should arrive in time to complete the daily update. Generally if you are on time or early, you will have time to complete the daily update before the deadline.

2.4.3 Readings (176 pts)

Readings (176 points): Each week for 11 weeks, two points are awarded for each ten minutes of reading, up to 16 points (80 minutes) per week.

We anticipate that you will actually spend about nine hours per week on activities related to this class, with three hours in class, maybe two hours in readings, and the other four hours in other forms of study, including taking practice tests and writing programs.

The first point is simply for reading. The second point is for making an oral report in class. As you read, you should prepare notes (talking points) about your readings. Prepare to talk for about three minutes. You can talk, for example, about what you studied, why you selected it, what you learned, and/or where you think your studies might lead you next.

For example, if you read for 50 minutes, that will earn you 5 points. If you then deliver a three-minute oral report on that reading, you will double those points.

Your study time must represent your undivided attention. (You cannot claim credit for reading while also watching TV, for instance.) It can be spent skimming a large number of pages or reading carefully a small number of pages. You can choose where to read.

2.4.4 Skills Tests (200 pts)

<http://byuh.doncolton.com/cs440/2151/sguide.pdf> is a study guide that includes specific details about the tests.

Conditional Probability (100 points) q45: We will study conditional probability for about two weeks and give you three chances on three separate days to get your best score on a 40-minute, 50-question probability test. Each question is worth 2 points.

The Conditional Probability test (q45) is offered on these days and times.

Thu Jan 15 CP, q45 Exam (10:10 to 10:50)

Thu Jan 22 CP, q45 Exam (10:10 to 10:50)

Thu Apr 16 Final, 10:00–12:50, GCB 101

Resolution (100 points) q41: We will study propositional calculus resolution for about two weeks and give you three chances on three separate days to get your best score on a 40-minute, 10-question resolution test. Each question is worth 10 points.

The Resolution test (q41) is offered on these days and times.

Thu Feb 19 Res Exam q41 (10:10 to 10:50)

Thu Feb 26 Res Exam q41 (10:10 to 10:50)

Thu Apr 16 Final, 10:00–12:50, GCB 101

These skills-based tests are based on skills taught in the book and in class. Practice tests are available after each test has been introduced, or by request.

2.4.5 Programming Labs (600 pts)

<http://byuh.doncolton.com/cs440/2151/sguide.pdf> is a study guide that includes specific details about the programming assignments, including how points are earned.

Vacuum Agent (200 points): You will program a vacuum agent. It will

compete with other vacuum agents at cleaning a (virtual) room. Your score will be based on your program's cleaning performance. This will be our focus for about three weeks.

Wumpus Agent (200 points): You will program a wumpus-hunting agent. It will compete with other agents. Your score will be based on your program's performance. This will be our focus for about three weeks.

Numbers Recognition (200 points): You will program a speech recognition program. It will receive phonetic transcriptions of spoken numbers and will convert them into words. You will compete with other such programs. Your score will be based on your program's performance. This will be our focus for about three weeks.

2.4.6 Final Exam Day

Final Exam: The final exam period will be an opportunity to retake either skills test and to get any of your programs retested.

I will calculate your final grade before final exam day, and tell you so you can decide whether to do any additional work.

Tests: If you want to take a skills test (or both of them), you must appear at the classroom at the time the final is scheduled to start. If nobody comes, I will shut down the tests.

Labs: Programming labs can be submitted in the normal way (by copying to a designated folder) and must be announced by also sending me an email. These do not require you to come to the classroom. The deadline for submitting labs is the start of the final exam period, Thu, Apr 16, 10:00–12:50.

2.4.7 Extra Credit

Report an error in my formal communications (the published materials I provide), so I can fix it. In this class, the materials include the following:

- The course website, parts relating to this semester.
- The course syllabus.
- The course study guide.

Each error reported can earn you extra credit. (Typos in my informal communications are all too common and do not count.)

Syllabus errors (unless they are major) will probably be fixed only in the study guide. Check there before reporting it.

2.5 Instructional Methods

My instructional methods match my grading plan, and I refer you to the grading section for more details. But in brief, there are four main methods that I use.

Readings: These expose students to the field of AI and require them to understand well enough to talk about what they read. See section [2.4.3](#) (page [13](#)).

Skills Tests: These expose students to some required mathematical proficiencies. See section [2.4.4](#) (page [14](#)).

Programming Activities: These require students to synthesize working programs based on knowledge we develop in class. See section [2.4.5](#) (page [14](#)).

Lecture: This happens as a side-effect of the other methods, especially your oral reports on your readings. Some days I spend the whole class period lecturing about a current topic. I review material that was assigned from the text book or study guide and do what I can to make it clear and interesting. This happens most often when I introduce a new exam or lab.

2.5.1 BYUH Learning Framework

I agree with the BYUH Framework for Learning. If we follow it, class will be better for everyone.

2.5.2 Prepare for CS 490R

Prepare: Before class, study the course material and develop a solid understanding of it. Try to construct an understanding of the big picture and how each of the ideas and concepts relate to each other. Where appropriate use study groups to improve your and others' understanding of the material.

In CS 490R: Read, practice, and program. Find interesting topics to tell or ask about.

2.5.3 Engage in CS 490R

Engage: When attending class actively participate in discussions and ask questions. Test your ideas out with others and be open to their ideas and insights as well. As you leave class ask yourself, “Was class better because I was there today?”

In CS 490R: Report on your readings and help us discuss the reports made by other students.

2.5.4 Improve at CS 490R

Improve: Reflect on learning experiences and allow them to shape you into a more complete person: be willing to change your position or perspective on a certain subject. Take new risks and seek further opportunities to learn.

In CS 490R: Read, practice, and program. Find interesting topics to tell or ask about.

2.5.5 Support

The major forms of support are (a) open lab, (b) study groups, and (c) tutoring.

If you still need help, please find me, even outside my posted office hours.

2.5.6 Office Hour / Open Lab

As mentioned above, I have reserved GCB 111 on MWF from 14:30 to 15:30 (right after my CIS 101 class) and on TuTh from 15:30 to 17:00 (right after my IT 240 class) so my students (and others) can study in a lab setting and meet with me and each other. I will be there at the start of those hours, and will stay as long as students are asking me questions. I also allow the room as an Open Lab for your use either individually or in groups, for my class or for other classes.

2.5.7 Study Groups

You are encouraged to form a study group. If you are smart, being in a study group will give you the opportunity to assist others. By assisting others you will be exposed to ideas and approaches (and errors) that you might never have considered on your own. You will benefit.

One possible time for your study group to meet is during the open lab time.

If you are struggling, being in a study group will give you the opportunity to ask questions from someone that remembers what it is like to be totally new at this subject. They are more likely to understand your questions because they sat through the same classes you did, took the same tests as you did, and probably thought about the same questions that you did.

Most of us are smart some of the time, and struggling some of the time. Study groups are good.

2.5.8 Tutoring

For a 400-level class, tutoring basically does not exist. Your best bet is to ask about things during class. You are also welcome to bring your questions to me during office hours or outside of office hours. And other faculty may enjoy discussing your questions with you.

2.6 Course Policies

Subject to Change: Like all courses I teach, I will be keeping an eye out for ways this one could be improved. Changes generally take the form of opportunities for extra credit, so nobody gets hurt and some people may be helped. If I make a change to the course and it seems unfair to you, let me know and I will try to correct it. If you still think it is unfair, you can appeal to the department chair or the dean. Also, you are welcome to suggest ways you think the class could be improved.

Digital Recording by me: I may digitally record the audio of my lectures some days. This is to help me improve my teaching materials.

Digital Recording by you: Almost everyone has a smart phone these days. I assume students will freely record what goes on in class, and take pictures of what is on the white board, to aid in their studies. I simply ask

that you not embarrass anyone.

2.6.1 Special Treatment

There are many good reasons why students request special treatment. These include, for example, illness, field trips, performances, athletic events, and special needs. Instead of dealing with these as they arise, based on my past experience, I have adopted general policies that are intended to accommodate all but the most difficult cases, and thereby avoid the need for special treatment.

2.6.2 Reasonable Accommodation

This section covers special needs, including qualified special needs, as well as all other requests for special treatment.

I have carefully designed each of my classes to provide what I believe to be reasonable accommodation to those with special needs. Beyond that, further accommodation is usually considered to be unreasonable but may happen in extreme cases. Please see the paragraph on “Accommodating Special Needs” below for more information.

Ample Time: Specifically, I allow ample time on tests so that a well-prepared student can typically finish each test in half of the time allowed. This gives everyone essentially double the amount of time that should normally be needed.

Exam Retakes: Each test is given three times (sometimes more), and I keep the highest score that was earned. This handles the case of persons that are unable to attend class or function at their best on any given day.

I consider the first attempt to be routine. I consider the second and third attempts to be an accommodation for anyone that might need it. The scheduled final exam is Thu, Apr 16, 10:00–12:50. The final consists of that third opportunity to retake **any** exam that was offered during the semester. If you are happy enough with your previous scores, **you can probably skip the final.**

As a side effect of this three-tries approach to tests, it is also true that any student can miss any one or two days of class for any reason without messing up their grade.

On the other hand, the retakes are limited. If you have issues every single time the test is given, I do draw the line, and I will not give additional chances. Additional retakes are not considered to be a “reasonable” accommodation. Additional retakes or make-ups are not offered except in highly unusual circumstances.

Deadlines: These are listed in the syllabus and in the study guide.

Even though I truly believe that these methods provide reasonable accommodation for almost everyone in almost every case, you might have a highly unusual situation for which I can and should do even more. You are welcome to see me about your situation.

2.6.3 Communication

We communicate with each other both formally and informally.

Formal communication is official, carefully worded, and normally in writing. Formal is for anything truly important, like grades.

Informal communication is casual and impromptu. It is meant to be helpful and efficient. Reminders are informal. Emails are informal. Explanations are usually informal.

From Me to You, Formal

I communicate formally, in writing, through (a) the syllabus, (b) the study guide, and (c) the learning management system.

(a) Syllabus: <http://byuh.doncolton.com/cs440/2151/syl.pdf> is the syllabus for this course. It tells our learning objectives and how you will be graded overall. You can rely on the syllabus. After class begins, it is almost never changed except to fix major errors.

(b) Study Guide: <http://byuh.doncolton.com/cs440/2151/sguide.pdf> is the study guide for this course. It includes a copy of the syllabus. The study guide is updated frequently throughout the semester, as assignments are made and deadlines are established or updated.

(b1) Calendar: The study guide tells when things will happen. It contains specific due dates.

(b2) Assignments: The study guide tells what assignments have been made and how you will be graded, item by item. It provides current details and specific helps for each assignment. It provides guidance for taking the tests.

(c) DCQuiz: <https://dcquiz.byuh.edu/> is my learning management system. I use it to give tests. I use it to show you my grade books.

From Me to You, Informal

My main informal channels to you are (a) word of mouth and (b) email.

(a) Word of Mouth, including Lecture: Class time is meant to be informative and helpful. But if I say anything truly crucial, I will also put it into the study guide.

(b) Email: My emails to you are meant to be helpful. But if I say anything truly crucial, I will also put it into the study guide. Normally I put CS 490R at the front of the subject line in each email I send.

From You to Me, Formal

Your formal channels to me, specifically how you turn in class work, are mainly via (a) the learning management system, (b) email, and (c) specifically requested projects.

(a) DCQuiz: To use my learning management system, you must log into it. Then, you can respond to questions I have posted. Each day there will be a “daily update”. I say more on that below. Tests will also be given using DCQuiz.

(b) Email: You will use formal email messages to submit some of the programs you write and to tell me certain other things. The study guide tells how to send formal emails, including where to send them, what subject line to use, and what to put in the body of the message.

(c) Student Projects: The study guide may tell you to submit certain work in the form of a webpage or web-based program. If so, it will say specifically where to put it. I will go to that spot to grade it.

From You to Me, Informal

Your informal channels to me, typically how you ask questions and get assistance, are mainly face to face and by email or chat.

Face to Face: If you need help with your class work, I am happy to look at it and offer assistance. Often this happens during class or during office hours. Often I will have you put your work on your computer screen, and then I will take a look at it while we talk face to face.

Email / Chat: You can also get assistance by sending me an email or doing a chat. I will do my best to respond to it in a reasonable and helpful way. If you want something formal, use the formal rules.

If you are writing about several different things you will usually get a faster response if you break it up into several smaller emails instead of one big email. I try to respond to a whole email at once, and not just part of it. I usually answer smaller and simpler emails faster than big ones.

2.7 Learning Outcomes

Outcomes (sometimes called objectives) are stated at several levels: ILO, PLO, and CLO. In this section we set forward these outcomes and tell how they are aligned with one another.

2.7.1 ILOs: Institutional Outcomes

ILO: Institutional Learning Outcomes (ILOs) summarize the goals and outcomes for all graduates of BYUH.

Brigham Young University Institutional Learning Objectives (ILOs) Revised
24 February 2014

Graduates of Brigham Young University–Hawai‘i will:

Knowledge: Have a breadth of knowledge typically gained through general education and religious educations, and will have a depth of knowledge in their particular discipline.

Inquiry: Demonstrate information literacy and critical thinking to understand, use, and evaluate evidence and sources.

Analysis: Use critical thinking to analyze arguments, solve problems, and

reason quantitatively.

Communication: Communicate effectively in both written and oral form, with integrity, good logic, and appropriate evidence.

Integrity: Integrate spiritual and secular learning and behave ethically.

Stewardship: Use knowledge, reasoning, and research to take responsibility for and make wise decisions about the use of resources.

Service: Use knowledge, reasoning, and research to solve problems and serve others.

2.7.2 PLOs: Program Outcomes

PLO: Program Learning Outcomes (PLOs) summarize the goals and outcomes for graduates in programs for which this course is a requirement or an elective. These support the ILOs, but are more specific.

At the end of this section, we include the relevant page from the CIS Program Outcomes Matrix, dated April 2011.

The following outcomes are pursued at the levels indicated.

High: **Demonstrated at Mastery Level**

(a) An ability to apply knowledge of computing and mathematics appropriate to the discipline.

High: **Demonstrated at Mastery Level**

(b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.

High: **Demonstrated at Mastery Level**

(c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs.

High: **Demonstrated at Mastery Level**

(h) Recognition of the need for and an ability to engage in continuing professional development.

High: **Demonstrated at Mastery Level**

(CS j) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.

High: **Demonstrated at Mastery Level**

(CS k) An ability to apply design and development principles in the construction of software systems of varying complexity.

2.7.3 CLOs: Course Outcomes

CLO: Course Learning Outcomes (CLOs, also called Student Learning Outcomes, or SLOs) summarize the goals and outcomes for students who successfully complete this course. These support the PLOs, but are more specific.

In this class we will gain the basic understanding of AI that is needed by every computer scientist. We will write several programs that utilize AI technology.

Course Goals and Student Learning Outcomes are as follows. By the conclusion of this course, students will do the following.

1. Explain fundamental issues in intelligent systems.
2. Explain search in terms of problem space and goal state.
3. Correctly perform propositional calculus resolution.
4. Correctly perform conditional probability calculations.
5. Explain and use Bayes theorem.
6. Construct and test intelligent agents in several settings.
7. Use a corpus to develop a speech recognition program.
8. Defend the need for having established corpora for speech.

2.8 General Topics

All syllabi are encouraged or required to address certain topics. These are generally considered to be common sense, but we find that it is useful to mention them explicitly anyway.

2.8.1 Academic Integrity

Applicable Actions

<http://honorcode.byuh.edu/> details the university honor code. In the section entitled “Applicable Actions” the following are listed.

Examples of possible actions include but are not limited to the following, for instructors, programs, departments, and colleges:

- Reprimanding the student orally or in writing.
- Requiring work affected by the academic dishonesty to be redone.
- Administering a lower or failing grade on the affected assignment, test, or course.
- Removing the student from the course.
- Recommending probation, suspension, or dismissal.

Depending on the specifics of the offense, any of these responses may be possible.

Cheating on exams is the most common form of dishonesty that I normally encounter. Normally this happens when students bring in notes that include answers to past exam questions. I approve the studying of past exams, and bringing in of “memories” based on study, but not the access to written notes, including notes retrieved from other exams or stored on cell phones or other devices. Any such activity, if caught, can result in failure of the entire course.

Cheating on activities is almost impossible to detect because I allow students to collaborate and assist each other. Copy and paste is not allowed, but it is difficult to detect and prove, so I normally do not bother. You should try to understand the work you submit because it helps you prepare for the exams and future courses.

Plagiarism

We learn by watching others and then doing something similar.

Plagiarism: Sometimes it is said that plagiarism is copying from one person, and research is “copying” from lots of people.

When you are having trouble with an assignment, I encourage you to look at not just one, but many examples of work done by others. Study the examples. See what you can learn from them. Do not automatically trust that they are right. They may be wrong.

Do not just copy. Do your own work. When I review computer code, sometimes I see quirky ways of doing things. They appear to work even

though they may be wrong. And then I see someone else that has done it exactly the same wrong way. This does not feel like “doing your own work.” Cut and paste is pretty much an honor code violation. Read and learn is totally okay. Copying other ideas is okay. I don’t want to see any cut and paste.

<http://en.wikipedia.org/wiki/Plagiarism> has a wonderful article on plagiarism. Read it if you are not familiar with the term. Essentially, plagiarism is when you present the intellectual work of other people as though it were your own. This may happen by cut-and-paste from a website, or by group work on homework. In some cases, plagiarism may also create a violation of copyright law. If you borrow wording from someone else, identify the source.

Intentional plagiarism is a form of intellectual theft that violates widely recognized principles of academic integrity as well as the Honor Code. Such plagiarism may subject the student to appropriate disciplinary action administered through the university Honor Code Office, in addition to academic sanctions that may be applied by an instructor.

Inadvertent plagiarism, whereas not in violation of the Honor Code, is nevertheless a form of intellectual carelessness that is unacceptable in the academic community. Plagiarism of any kind is completely contrary to the established practices of higher education, where all members of the university are expected to acknowledge the original intellectual work of others that is included in one’s own work.

CS 490R: In this course group work is permitted and encouraged but you are not allowed to turn in work that is beyond your understanding, whether you give proper attribution or not. Make sure you understand what you are submitting and why each line is there.

CS 490R: On exams you are required to work from personal memory, using only the resources that are normally present on your computer. This means the exams are closed book and closed notes. However, you are nearly always allowed (and encouraged!) to test your program by actually running it on the computer where you are sitting. Students caught cheating on the final exam may receive a grade of F for the semester, no matter how many points they may have earned, and they will be reported to the Honor Code office.

Faculty are responsible to establish and communicate to students their expectations of behavior with respect to academic honesty and student conduct in the course. Observations and reports of academic dishonesty shall be investigated by the instructor, who will determine and take appropriate action, and report to the Honor Code Office the final disposition of any incident of academic dishonesty by completing an Academic Dishonesty Student Violation Report. If the incident of academic dishonesty involves the violation of a public law, e.g., breaking and entering into an office or stealing an examination, the act should also be reported to University Police. If an affected student disagrees with the determination or action and is unable to resolve the matter to the mutual satisfaction of the student and the instructor, the student may have the matter reviewed through the university's grievance process.

2.8.2 Sexual Misconduct

Sexual Harassment is unwelcome speech or conduct of a sexual nature and includes unwelcome sexual advances, requests for sexual favors, and other verbal, nonverbal, or physical conduct. Conduct is unwelcome if the individual toward whom it is directed did not request or invite it and regarded the conduct as undesirable or offensive.

Brigham Young University–Hawai'i is committed to a policy of nondiscrimination on the basis of race, color, sex (including pregnancy), religion, national origin, ancestry, age, disability, genetic information, or veteran status in admissions, employment, or in any of its educational programs or activities.

University policy and Title IX of the Education Amendments of 1972 prohibits sexual harassment and other forms of sex discrimination against any participant in an educational program or activity at BYUH, including student-to-student sexual harassment.

The following individual has been designated to handle reports of sexual harassment and other inquiries regarding BYUH compliance with Title IX:

Debbie Hippolite-Wright
Title IX Coordinator
Vice President, Student Development & Life
Lorenzo Snow Administration Building
55-220 Kulanui Street

Laie, Hawaii 96762
Office Phone: 808-675-4819
E-Mail: debbie.hippolite.wright@byuh.edu
Sexual Harassment Hotline: 808-780-8875

BYUH's Office of Honor upholds a standard which states that parties can only engage in sexual activity freely within the legal bonds of marriage between a man and a woman. Consensual sexual activity outside the bonds of marriage is against the Honor Code and may result in probation, suspension, or dismissal from the University.

2.8.3 Dress and Grooming Standards

The dress and grooming of both men and women should always be modest, neat and clean, consistent with the dignity adherent to representing The Church of Jesus Christ of Latter-day Saints and any of its institutions of higher learning. Modesty and cleanliness are important values that reflect personal dignity and integrity, through which students, staff, and faculty represent the principles and standards of the Church. Members of the BYUH community commit themselves to observe these standards, which reflect the direction given by the Board of Trustees and the Church publication, "For the Strength of Youth." The Dress and Grooming Standards are as follows:

Men. A clean and neat appearance should be maintained. Shorts must cover the knee. Hair should be clean and neat, avoiding extreme styles or colors, and trimmed above the collar leaving the ear uncovered. Sideburns should not extend below the earlobe. If worn, moustaches should be neatly trimmed and may not extend beyond or below the corners of mouth. Men are expected to be clean shaven and beards are not acceptable. (If you have an exception, notify the instructor.) Earrings and other body piercing are not acceptable. For safety, footwear must be worn in all public places.

Women. A modest, clean and neat appearance should be maintained. Clothing is inappropriate when it is sleeveless, strapless, backless, or revealing, has slits above the knee, or is form fitting. Dresses, skirts, and shorts must cover the knee. Hairstyles should be clean and neat, avoiding extremes in styles and color. Excessive ear piercing and all other body piercing are not appropriate. For safety, footwear must be worn in all public places.

2.8.4 Accommodating Special Needs

Brigham Young University–Hawai‘i is committed to providing a working and learning atmosphere, which reasonably accommodates qualified persons with disabilities. If you have a disability and need accommodations, you may wish to self-identify by contacting:

Services for Students with Special Needs

McKay 181

Phone: 808-675-3518 or 808-675-3999

Email address: aunal@byuh.edu

The Coordinator for Students with Special Needs is Leilani A‘una.

Students with disabilities who are registered with the Special Needs Services should schedule an appointment with the instructor to discuss accommodations. If the student does not initiate this meeting, it is assumed no accommodations or modifications will be necessary to meet the requirements of this course. After registering with Services for Students with Special Needs, and with permission of the student, Letters of Accommodation will be sent to instructors.

If you need assistance or if you feel you have been unlawfully discriminated against on the basis of disability, you may seek resolution through established grievance policy and procedures. You should contact the Human Resource Services at 808-780-8875.

2.9 Syllabus Summary

Brigham Young University–Hawai‘i has adopted certain requirements relating to the information that must be provided in syllabi. This section lists those requirements and for each item either provides the information directly or gives a link to where it is provided above.

Course Information: See section [2.2.1](#) (page 9).

Title: Intelligent Systems

Number: CS 490R

Semester/Year: Winter 2015

Credits: 3

Prerequisites: CS 301

Location: GCB 101

Meeting Time: TuTh 09:20 to 10:50

Faculty Information: See section [2.2.2](#) (page 9).

Name: Don Colton

Office Location: GCB 128

Office Hours: MWF 14:30-15:00, TuTh 15:30-16:00.

Telephone: 808-675-3478

Email: doncolton2@gmail.com

Course Readings/Materials: See section [2.2.3](#) (page 10) for a list of textbooks, supplementary readings, and supplies required.

Course Description: See section [2.2.1](#) (page 9).

Expected Proficiencies:

See section [2.1.1](#) (page 9) for the proficiencies you should have before undertaking the course.

Course Goals and Student Learning Outcomes, including Alignment to Program (PLOs) and Institutional (ILOs) Learning Outcomes, and extent of coverage.

See section [2.7](#) (page 22) for learning outcomes, showing the content of the course and how it fits into the broader curriculum. A listing of the

departmental learning outcomes is provided together with the ratings taken from department's matrix assessment document representing the degree to which the course addresses each outcome.

Instructional Methods: See section 2.5 (page 16).

Learning Management System:

<https://dcquiz.byuh.edu/> is the learning management system for my courses.

Framework for Student Learning:

See section 2.5.1 (page 16) for a discussion of the student learning framework and how I use it.

Course Calendar: See section 2.3 (page 11) for the calendar in general.

Here are some items of particular interest:

First Day of Instruction: Tue, Jan 13

Last Day to Withdraw: Tue, Mar 10

Last Day of Instruction: Tue, Apr 14

Final Exam: Thu, Apr 16, 10:00–12:50

Final Exam Location: GCB 101

Course Policies: See section 2.6 (page 18).

Attendance: See section 2.4.2 (page 13).

Tardiness: See section 2.4.2 (page 13).

Class Participation: See section 2.5.3 (page 17).

Make-Up Exams: See section 2.6.2 (page 19).

Plagiarism: See section 2.8.1 (page 27).

Academic Integrity: See section 2.8.1 (page 26).

Evaluation (Grading): See section 2.4 (page 12).

Academic Honesty: See section 2.8.1 (page 26).

Sexual Harassment and Misconduct: See section 2.8.2 (page 29).

Grievances: The university grievance policy states that the policies listed on the syllabus can act as a contract and will be considered if a student complains about the faculty.

Services for Students with Special Needs: See section [2.8.4](#) (page 31).

Chapter 3

Readings

You are assigned to read in the textbook for a certain amount of time each week. However, your specific readings are self directed.

Our textbook, *Artificial Intelligence, A Modern Approach, Second Edition*, by: Stuart Russell and Peter Norvig (1080 pages. ISBN: 0-13-790395-2. Prentice Hall.) is a department rental. (The third edition has been published but we will continue to use the second edition because it is adequate for our needs.)

While reading, you should prepare notes (talking points) about your readings so you can report in class. Be prepared to talk for about three minutes over what you studied, why you selected it, what you learned, and where you think your studies might lead you next.

Chapter 12 (page 73) contains a preliminary list of topics that you should know about. Read through that list and if something catches your attention, make a note of it and follow up in your readings.

Appendix A (page 80) identifies learning objectives considered important by the curriculum committees established by the two largest professional societies in computing. Again, read through that list and if something catches your attention, make a note of it and follow up in your readings.

If you think you might like to study Artificial Intelligence / Intelligent Systems at graduate school, you are strongly encouraged to dig deeply into the textbook. It is excellent and will give you incredible preparation for that field.

Chapter 4

Conditional Probability

Contents

4.1 Monty Hall	37
4.2 Strategy: Venn	37
4.3 Strategy: Table	38
4.4 Notation	39
4.5 Sample Problems	40

- Exam #1: Thu, 2015-01-15 (10:10 to 10:50)
- Exam #2: Thu, 2015-01-22 (10:10 to 10:50)
- Exam #3: Thu, 2015-04-16 (10:00 to 12:50)

Probability, Conditional Probability, and Updating of Probability are important skills relating to Artificial Intelligence and Intelligent Systems.

This is true because intelligent systems may be required to select between different possible actions where the outcomes are not certain, but can be estimated in terms of **probability**. To make the best choice, expected values must be calculated.

http://en.wikipedia.org/wiki/Conditional_probability

For this class, we will test you by giving you probability information and asking you to calculate other probability information.

Probabilities are always numbers between zero (meaning it never happens)

and one (meaning it always happens). We will normally write them as fractions.

You are given certain probabilities. Your task is to find the requested probability. Express your answer as a reduced fraction (like $5/7$).

4.1 Monty Hall

To show how difficult it is to guess the right probability, consider the Monty Hall problem. (Do a Google search if necessary.)

Monty Hall, a famous US television host, offers you a prize behind one of three doors. Behind one door is a new car. Behind each other door is a goat. You have a 1-in-3 chance of picking the car. You pick a door.

After you pick a door, Monty opens one of the other two doors, revealing a goat.

Question: What is the probability that you picked the car, given that you know where one of the goats is? If you have the opportunity to switch your choice, should you do it?

Most people have a very hard time figuring this out. Is your probability still $1/3$? Is your probability now $1/2$? What do you believe?

Thinking about it another way, let's have 100 doors with 99 goats. After you pick a door, Monty opens 98 other doors, revealing 98 goats. Now there are two doors closed, the one you picked and the one he did not reveal. What is your probability now? Is it $1/100$? Is it $1/2$? Is it something else entirely?

4.2 Strategy: Venn

Most students are familiar with the **Venn diagram**.

http://en.wikipedia.org/wiki/Venn_diagram has a wonderful article that gives useful information. Even students familiar with Venn may learn something new.

One good approach to solving problems in conditional probability is to construct a Venn diagram using the facts at hand. Then, using the Venn diagram, determine the answer.

4.3 Strategy: Table

Another approach is to construct a three-by-three probability table.

	B	-B	
A	(1)	(2)	(3)
-A	(4)	(5)	(6)
	(7)	(8)	(9)

(1) is the joint probability that A and B are both true.

(2) is the joint probability that A is true and B is false.

(3) is the simple probability that A is true, whether or not B is true.
It is the sum of (1) and (2).

(4) is the joint probability that A is false and B is true.

(5) is the joint probability that A and B are both false.

(6) is the simple probability that A is false, whether or not B is true.
It is the sum of (4) and (5).

(7) is the simple probability that B is true, whether or not A is true.
It is the sum of (1) and (4).

(8) is the simple probability that B is false, whether or not A is true.
It is the sum of (2) and (5).

(9) is always 1.0.

It is the sum of (3) and (6).

It is also the sum of (7) and (8).

For simplicity, if we are working with whole numbers, we can write in (9) the denominator (least common multiple) that applies to all cells. That is, instead of writing $6/13$ in cell (1), we could write 6 in cell (1) and 13 in cell (9).

We begin by drawing the grid and labeling the sides. Then we consult our given information and fill in the blanks that we know.

If we know the probability that A is true, we write it in cell (3). We can then calculate the probability that A is false and write it in cell (6).

If we know the probability that B is true, we write it in cell (7). We can then calculate the probability that B is false and write it in cell (8).

Eventually we can read off the cells (2), (1), (4), and (5), to get the values that would be used in a Venn diagram.

4.4 Notation

You should be familiar with the commonly used notations relating to probability. These are often written with mathematical symbols.

http://en.wikipedia.org/wiki/Truth_table has much more.

Primitives: These are some of the primitive wordings and operations used with probability.

$p(x)$: When you see $p()$ read it as “the probability that ... is true”.

\cap : When you see \cap read it as the word “and”. It can also be read as the word “intersection”. It can be pronounced “cap”. It is also called **conjunction**.

\cup : When you see \cup read it as the word “or”. It can also be read as the word “union”. It can be pronounced “cup”. It is also called **disjunction**.

\bar{x} : When you see a bar over something, read it as the word “not”.

$|$: When you see $|$ read it as the word “given”.

\rightarrow : When you see \rightarrow read it as the word “implies”. Note that “implies” is not the same as “causes”.

Expressions: These are typical combinations of the five primitives into longer expressions.

$p(A)$ means “the probability that A is true”.

$p(A)=5/7$ means that in the universe of possibilities, there are basically seven equally likely groupings of things, and in five of them A is true.

$p(A \cap B)$ means the (joint) probability that both A and B are true. We may also write this as **p(A and B)**.

$p(\bar{B})$ means “the probability that not B is true”, or in other words, “the probability that B is false”. We may also write this as **p(not B)**.

$p(A \cap \bar{B})$ means the probability that A is true and B is false. We may also write this as **p(A and not B)**.

$p(A|B)$ means the probability that A is true if we already know that B is true. We may also write this as **p(A given B)**.

$p(A \rightarrow B)$ means the probability that if A is true, then B is also true. We may also write this as **p(A implies B)**.

4.5 Sample Problems

<http://quizgen.doncolton.com/> quiz q45 provides additional opportunities for you to learn and practice these skills.

Given $p(A)=1/3$, $p(B)=7/18$, $p(A \cap B)=1/9$.

You should first derive the following Venn diagram: (4(2)5)7.

The “(4(2)” part represents the A circle in the **Venn diagram**. It means that in four cases, A is true but B is not true. In two cases A is true and B is true. It does not say anything about when A is false.

The “(2)5)” part represents the B circle in the Venn diagram. It means that in two cases, B is true and A is also true. In five cases B is true but A is not true. It does not say anything about when B is false.

The “7” part represents the space outside the A and B circles. It means that in seven cases both A and B are false.

Each of these 4, 2, 5, and 7 cases are independent and equally likely, for a total of 18 possible cases.

Exam Question 2 (p.89): Find $p(A \cap \bar{B})$ given $p(A)=1/3$, $p(B)=7/18$, $p(A \cap B)=1/9$.

Acceptable Answer: 2/9

Exam Question 3 (p.89): Find $p(\bar{A} \cap B)$ given $p(A)=1/3$, $p(B)=7/18$, $p(A \cap B)=1/9$.

Acceptable Answer: 5/18

Exam Question 4 (p.90): Find $p(\bar{A} \cap \bar{B})$ given $p(A)=1/3$, $p(B)=7/18$, $p(A \cap B)=1/9$.

Acceptable Answer: 7/18

Exam Question 5 (p.90): Find $p(A|B)$ given $p(A)=1/3$, $p(B)=7/18$, $p(A \cap B)=1/9$.

Acceptable Answer: 2/7

Exam Question 6 (p.90): Find $p(A|\bar{B})$ given $p(A)=1/3$, $p(B)=7/18$, $p(A \cap B)=1/9$.

Acceptable Answer: $4/11$

Exam Question 7 (p.90): Find $p(B|A)$ given $p(A)=1/3$, $p(B)=7/18$, $p(A \cap B)=1/9$.

Acceptable Answer: $1/3$

Exam Question 8 (p.90): Find $p(B|\bar{A})$ given $p(A)=1/3$, $p(B)=7/18$, $p(A \cap B)=1/9$.

Acceptable Answer: $5/12$

Given $p(A)=5/6$, $p(B)=2/3$, $p(A \cap B)=7/12$, you should first derive the following Venn diagram: (3(7)1)1.

Exam Question 9 (p.90): Find $p(\bar{A} \cap B)$ given $p(A)=5/6$, $p(B)=2/3$, $p(A \cap B)=7/12$.

Acceptable Answer: $1/12$

Exam Question 10 (p.90): Find $p(\bar{A} \cap \bar{B})$ given $p(A)=5/6$, $p(B)=2/3$, $p(A \cap B)=7/12$.

Acceptable Answer: $1/12$

Exam Question 11 (p.90): Find $p(A|B)$ given $p(A)=5/6$, $p(B)=2/3$, $p(A \cap B)=7/12$.

Acceptable Answer: $7/8$

Exam Question 12 (p.90): Find $p(A|\bar{B})$ given $p(A)=5/6$, $p(B)=2/3$, $p(A \cap B)=7/12$.

Acceptable Answer: $3/4$

Exam Question 13 (p.90): Find $p(B|A)$ given $p(A)=5/6$, $p(B)=2/3$, $p(A \cap B)=7/12$.

Acceptable Answer: $7/10$

Exam Question 14 (p.90): Find $p(B|\bar{A})$ given $p(A)=5/6$, $p(B)=2/3$, $p(A \cap B)=7/12$.

Acceptable Answer: $1/2$

Given $p(A)=3/7$, $p(B)=3/7$, $p(A \cap B)=2/21$, you should first derive the following Venn diagram: (7(2)7)5.

Exam Question 15 (p.90): Find $p(A \cap \bar{B})$ given $p(A)=3/7$, $p(B)=3/7$, $p(A \cap B)=2/21$.

Acceptable Answer: $1/3$

Exam Question 16 (p.90): Find $p(\bar{A} \cap B)$ given $p(A)=3/7$, $p(B)=3/7$, $p(A \cap B)=2/21$.

Acceptable Answer: $1/3$

Exam Question 17 (p.90): Find $p(\overline{A} \cap \overline{B})$ given $p(A)=3/7$, $p(B)=3/7$,
 $p(A \cap B)=2/21$.

Acceptable Answer: $5/21$

Exam Question 18 (p.90): Find $p(A|B)$ given $p(A)=3/7$, $p(B)=3/7$,
 $p(A \cap B)=2/21$.

Acceptable Answer: $2/9$

Exam Question 19 (p.90): Find $p(A|\overline{B})$ given $p(A)=3/7$, $p(B)=3/7$,
 $p(A \cap B)=2/21$.

Acceptable Answer: $7/12$

Exam Question 20 (p.90): Find $p(B|A)$ given $p(A)=3/7$, $p(B)=3/7$,
 $p(A \cap B)=2/21$.

Acceptable Answer: $2/9$

Exam Question 21 (p.90): Find $p(B|\overline{A})$ given $p(A)=3/7$, $p(B)=3/7$,
 $p(A \cap B)=2/21$.

Acceptable Answer: $7/12$

Chapter 5

Bayesian Probability

Contents

5.1 The Product Rule	43
5.2 Detailed Example	44
5.3 Bayes' Rule	45

http://en.wikipedia.org/wiki/Thomas_Bayes has information about Thomas Bayes (1702-1761).

Bayes comes up a lot in AI in connection with probabilistic inference. It is surprisingly simple.

5.1 The Product Rule

The **product rule** is this:

$$p(a \wedge b) = p(a|b)p(b) \text{ or } p(a \wedge b) = p(b|a)p(a)$$

We will show an example of this shortly.

Combining these two forms, we get **Bayes' rule**, aka Bayes' law, aka Bayes' theorem, which is this:

$$p(b|a) = p(a|b)p(b)/p(a)$$

(For some reason, I have a hard time remembering the Bayes' rule, but a somewhat easier time remembering the product rule. Fortunately for me, it is easy to derive Bayes from product.)

Each of $p(a)$ and $p(b)$ are called **prior probabilities**, or **a priori probabilities**. They are the probabilities that something is true when you don't know anything else about anything.

$p(a|b)$ is the conditional probability that a is true given that you already know b is true.

Exam Question 22 (p.90): What is the product rule?

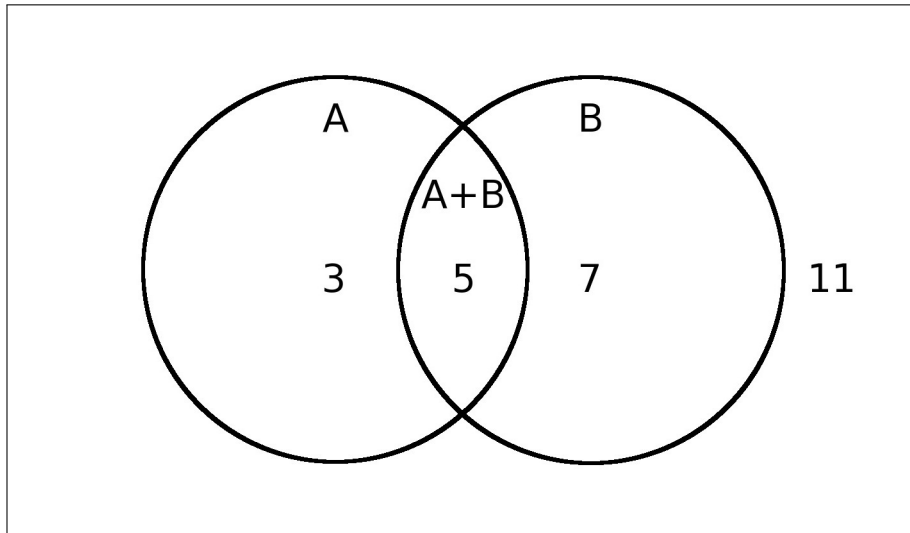
Acceptable Answer: $p(a \text{ and } b) = p(a \text{ given } b)p(b)$

Exam Question 23 (p.90): What is Bayes' rule?

Acceptable Answer: $p(a \text{ given } b) = p(b \text{ given } a)p(a)/p(b)$

5.2 Detailed Example

Let's look at what this means by way of a Venn diagram.



In this example, $p(A) = 8/26$, $p(A \wedge B) = 5/26$, and $p(B) = 12/26$.

Bayes' rule is used to calculate $p(b|a)$ when our basic facts include $p(a|b)$, $p(a)$, and $p(b)$. We do this as follows.

We know that $p(A \wedge B) = p(A|B)p(B)$. What does that mean? $p(A|B)$ is the probability of A given that B is known to be true.

In this case, we have $5+7=12$ cases where B is true, and in 5 of them, A is also true. Hence, $p(A|B)$ is $5/12$.

We now have our three facts: $p(A|B) = 5/12$, $p(A) = 8/26$, and $p(B) = 12/26$.

The product rule tells us that multiplying $p(A|B)$ by $p(B)$ we get $p(A \wedge B)$. Specifically, $5/12$ times $12/26$ equals $5/26$. Notice that the 12s cancel each other out.

The product rule tells us that dividing $p(A \wedge B)$ by $p(A)$ we get $p(B|A)$ which is our goal. $5/26$ divided by $8/26$ equals $5/8$. Notice that the 26s cancel each other out.

We can check our work by counting up the chances directly. We see there are 8 chances for a to be true, and in 5 of them, b is also true. Therefore $p(b|a)$ is $5/8$.

Bayes' rule just combines the two applications of the product rule.

$$p(B|A) = p(A|B)p(B)/p(A)$$

$$p(B|A) = (5/12)(12/26)/(8/26)$$

5.3 Bayes' Rule

Let's look at two propositions.

A: The sound ah was uttered.

B: The computer thinks it heard the sound ah.

We want to know the probability that ah was uttered when the computer recognizes an ah: $p(u|r)$.

We can find the prior probability that ah was **uttered** by looking at a corpus of speech labeled by trained human transcribers. Let's make up a number and say that out of some corpus of speech, ah is uttered in one percent of the frames.

$$p(u) = 0.01$$

We can find the prior probability that ah was **recognized** by looking at a similar corpus of speech labeled by computer. Let's make up a number and say that in that corpus of speech, ah is recognized in two percent of the frames.

$$p(r) = 0.02$$

Due to confusion between similar phonemes, let's say that out of the times

ah was actually uttered, it is recognized as the most likely phoneme 3/4 of the time.

$$p(r|u) = 0.75$$

From this we can calculate $p(u|r)$, the probability that ah was uttered given the computer recognized it.

$$p(u|r) = p(r|u)p(u)/p(r) = 0.75 * 0.01 / 0.02 = 0.375.$$

I guess in this case our computer is not very accurate yet.

Chapter 6

Propositional Resolution

Contents

6.1	Notation	48
6.2	How to Resolve	49
6.3	Canonical Grading Form	50
6.4	Sample Problems	50

- Exam #1: Thu, 2015-02-19 (10:10 to 10:50)
- Exam #2: Thu, 2015-02-26 (10:10 to 10:50)
- Exam #3: Thu, 2015-04-16 (10:00 to 12:50)

Resolution is the process of combining truth clauses to arrive at the most simplified version of those clauses.

We will restrict our attention to the domain called **Propositional Logic** or “Propositional Calculus”. (There is another more complicated domain called **Predicate Calculus** or “first order predicate calculus” or simply **first order logic**.) In the propositional domain, the basic (atomic) clauses are simple propositions that have a well-defined truth value, being either true or false.

For example, the clauses “It is raining” and “If it is raining then the ground will be wet” can be used to deduce that “The ground will be wet.”

Many years ago philosophers and logicians created lists of rules by which these deductions could be made. They are called **rules of inference**, or

sometimes **sylogisms**. Many of them have Latin names, like **modus ponens**.

http://en.wikipedia.org/wiki/Modus_ponens has an introduction to the subject, and includes a list of other rules of inference in propositional calculus and predicate calculus.

In 1965 the philosopher / mathematician / computer scientist **John Alan Robinson** pointed out that all these rules of inference can be simplified down to a reliable and complete process called **resolution**.

[http://en.wikipedia.org/wiki/Resolution_\(logic\)](http://en.wikipedia.org/wiki/Resolution_(logic)) discusses resolution.

We expect you to learn and correctly apply resolution to a series of clauses that you will be given. We explain how to do that below.

6.1 Notation

Vocabulary: The following words are important.

Conjunction means “and” and is sometimes represented by the symbol \wedge .

Disjunction means “or” and is sometimes represented by the symbol \vee .

Basics: (1) The list of TRUE statements is called the knowledge base (the KB). We will add and delete statements to improve the list. (2) Each statement is called a clause, and consists of an or-list (**disjunction**) of simple propositions. If it is in the KB, it is claimed to be TRUE. (3) Each simple proposition is either TRUE or FALSE. Put another way, for every proposition (p), the clause “(p) or (not p)” must be TRUE.

“I am dry” is a simple proposition, and “I have an umbrella” is a simple proposition. Typically we abbreviate propositions down to letters or short words for convenience. The statement “I am dry, or I don’t have an umbrella” could be written as “(dry) or (not umbrella)” or “dry -umb” or even “d -u”. Each such statement is called a clause. As mentioned above, every clause in the KB is asserted to be TRUE.

We will express our knowledge base as a **conjunction** of **disjunctions**.

http://en.wikipedia.org/wiki/Conjunctive_normal_form discusses this form, conjunctive normal form, which is also called **CNF**.

Each clause is a **disjunction** of propositions. In a clause expressed as a **disjunction** of propositions, one of the propositions must be true. If one

proposition is true, then the whole clause becomes true.

Each clause in the knowledge base must be true. When all must be true, we call that a **conjunction**.

6.2 How to Resolve

Rule 1 (clause simplification): We can drop any proposition we know to be FALSE from a clause because the whole clause must be TRUE, and the FALSE part clearly isn't helping. This is exactly like regular math where adding zero does not change anything. If $x+0=5$ we can drop the $+0$ and be left with $x=5$. If (a) or (b) or (c) is TRUE, and we know that (a) is FALSE, then we can drop it, leaving (b) or (c) is TRUE.

Rule 2 (reduction): Adding more propositions to a clause (an or-list) cannot change it from TRUE to FALSE. That is the nature of OR. If we know (b) is TRUE, then (b) or (c) is also TRUE, no matter what (c) is. As a direct result, if we have two clauses in the KB, and the little one is an exact subset of the big one, we can throw away the big one without losing any information. For example, if we know (i can drive), it does not help to also say (i can drive) or (i am rich). It tells nothing about my riches.

Rule 2 (special case): If we have a clause that includes “a -a” within it, we can delete it from the KB. The clause provides no information, since we already know that either (a) or (not a) is TRUE. It is called a **tautology**, and the “a” and “-a” are called **complementary literals**.

Rule 3 (resolution): New clauses can be created from old clauses. This part is a little tricky, but it is very powerful. If we have the clause “a x”, and we also have the clause “-a y”, we can combine them to create a new clause. Notice that one includes “a” in its or-list, and the other includes the opposite, “-a”. By basic rule 1, (a) is either TRUE or FALSE. Here is the tricky part. IF (a) is TRUE, then (not a) must be FALSE, so the second clause simplifies into (y) is TRUE. On the other hand, if (a) is FALSE, then the first clause simplifies into (x) is TRUE. We can splice the original clauses together to deduce “x y”. More generally, if we have (lots of things) or (a), and (other things) or (not a), we can combine them into (lots of things) or (other things). That's resolution.

Rule 3 (gotcha): If you have “a b more1” and “-a -b more2” there is a temptation to conclude “more1 more2”. However, one cannot combine on

both “a” and “b” at the same time. The correct combination on “a” would result in “b more1 -b more2” which includes “b -b” which is always true. Hence by the special case already mentioned, the conclusion can be dropped.

Procedure: To simplify the KB, we look at pairs of clauses. If one is a subset of the other, we delete the bigger one. If they share complementary propositions (one has (a) and the other has (not a)) we use resolution to generate a new clause and insert it into the KB (unless the new clause was already there). We continue looking at all pairs of clauses until no more insertions or deletions can be made. Careful ordering can make the job much faster; always reduce when you can. But whatever order you do things, you will always get the same result in the end.

6.3 Canonical Grading Form

There are many ways to write the same thing. The clauses could potentially be in any order. The propositions within the clauses could potentially be in any order.

For ease in grading, we require the following order. This makes all correct answers look the same, so grading can be done by exact match. (Putting it in this order is easier than it sounds.)

- (1) First arrange each clause into alphabetical order. The sorting order is - (dash) then [a-z].
- (2) Have a single space before and after each proposition. This makes it easier for us to catch mistakes before grading occurs.
- (3) Next order the whole clauses into alphabetical order.
- (4) Have a single space between clauses.

Everything should fit onto a single line, a **conjunction of disjunctions, conjunctive normal form (CNF)**.

Additional lines, such as comments, may not be included.

6.4 Sample Problems

Reduce each of the following to its most simple form through resolution. Each letter represents a proposition that is either TRUE or FALSE. Each

(clause) is a TRUE statement (at least one proposition is TRUE).

<http://quizgen.doncolton.com/> quiz q41 provides additional opportunities for you to learn and practice these skills.

Exam Question 24 (p.91): Resolve: $(-a c) (-b a) (-c a) (-c b) (b c)$
Acceptable Answer: $(a) (b) (c)$

Exam Question 25 (p.91): Resolve: $(-a b) (-b c) (a b)$
Acceptable Answer: $(b) (c)$

Exam Question 26 (p.91): Resolve: $(-c -d a) (a b d) (a c) (a c d)$
Acceptable Answer: $(-d a) (a b) (a c)$

Exam Question 27 (p.91): Resolve: $(-b -c a) (-c b d) (-d a b)$
Acceptable Answer: $(-c a) (-c b d) (-d a b)$

Exam Question 28 (p.91): Resolve: $(-a d) (-b c) (-c a) (-d a c) (a b)$
Acceptable Answer: $(-b c) (a) (d)$

Exam Question 29 (p.91): Resolve: $(-b a) (-b a c) (-c -d a) (a b c)$
Acceptable Answer: $(-b a) (-d a) (a c)$

Exam Question 30 (p.91): Resolve: $(-a -b c) (-b -c d) (-b a)$
Acceptable Answer: $(-b a) (-b c) (-b d)$

Exam Question 31 (p.91): Resolve: $(-a -c b) (-a -d c) (-b d) (-c a b)$
Acceptable Answer: $(-a -b c) (-a -d b) (-a -d c) (-b d) (-c b) (-c d)$

Exam Question 32 (p.91): Resolve: $(-c -d a) (-d b c) (a b) (a c d) (c d)$
Acceptable Answer: $(-c -d a) (a b) (b c) (c d)$

Exam Question 33 (p.91): Resolve: $(-a -d c) (-d a) (a b)$
Acceptable Answer: $(-d a) (-d c) (a b)$

Chapter 7

Projects in General

Contents

7.1 How To Submit	52
7.2 Choice of Language	53
7.3 Bake Off	54
7.4 Drivers	54

A substantial portion of your learning in this class will come as a result of programs that you write.

7.1 How To Submit

You will submit your projects by copying the files into a special inbox on the IS2 machine. The location of the inbox is:

```
~dc/inbox/
```

The inbox is set up as a “write only” destination. That means you can copy things into the inbox but you cannot see them or edit them or retrieve them from the inbox.

To submit your work, first, create or import the files into your own file space on the IS2 machine. Make sure the files are complete and work properly.

Next, copy (do not move) the files into the inbox. For example, if your file is named “John1”, you would copy it using the following command:

```
cp John1 ~dc/inbox/
```

Note: “mv” also works, kind of, but sometimes fails. “cp” is more reliable.

Note: Some students have attempted to upload their files directly into the inbox. Very often this fails because the inbox is “write only.” It is much more reliable to upload the files into your own file space first and then to copy them to the inbox.

Note: Some students have attempted to create their files directly within the inbox by using a text editor. Very often this fails because the inbox is “write only.” It is much more reliable to create the files into your own file space first and then to copy them to the inbox.

7.2 Choice of Language

You can use any programming language that supports these features:

- (a) Programs will be run under Linux on the IS2 machine.
- (b) Your program can use the name assigned by the instructor. Typically this consists of your own name followed by one or more digits. The first version of my program might be assigned the name “Don1” and the second version “Don2”.
- (c) Your program must successfully run from the command line by typing its assigned name, such as “./Don1”. There will be no command line arguments.
- (d) Your program can receive and process character input provided through STDIN.
- (e) Your program can provide character output through STDOUT.
- (f) Your program is permitted to have and to create additional files, but all additional files must start with the assigned name followed by a dot. If the name is “yadda” (for example), your main executable file must be named “yadda” and you may have additional files named “yadda.*” where the “*” can be anything, including a directory (folder) within which your files can be named anything you like.

C and C++ are known to work well.

Perl is known to work well.

Ruby is known to work well.

Tcl/Expect is known to work well.

Java is known to work, but it's a bit more tricky. You will probably use a “wrapper” for your primary executable, and it will start your actual program. Also, Java seems to have very long start-up times which can make your program appear to run slowly.

7.3 Bake Off

We will compare student programs (projects) by competing in a **bake off**. This terminology is commonly used and has reference to contests of cooking skills where each baker is required to produce some item of food that will be judged. By judging the food item we assess the skills of the baker.

Essentially we will give each program the same starting point and then run them simultaneously. Programs will earn points according to the rules of each project. Those with the most points will receive the better grades.

You will have access to the testing harness under which your program will be judged. This will allow you to test your program before submitting it for grading.

7.4 Drivers

Each project has an associated driver: the Vacuum Driver, the Wumpus Driver, and the Numbers Driver. These can be downloaded from the course homepage. There may be adjustments to the drivers while the projects are going on. If that happens, I will let you know in class. But you can always get the latest drivers from the webpage.

Chapter 8

Vacuum

Contents

8.1	Vacuum Rules	55
8.2	Vacuum Scoring	57
8.3	Vacuum Agent	58

- Bake Off #1: Thu, 2015-01-29, in class.
- Bake Off #2: Thu, 2015-02-05, in class.
- Bake Off #3: Thu, 2015-02-12, in class.
- Bake Off #4: Thu, 2015-04-16, by 10:00.

For this task, you will program a **vacuum** cleaner. You should visit every place in the room, clean what is dirty, return to your home base, and quit.

The room is divided into a grid of squares. Each square is either occupied by an obstacle, or is dirty, or is clean. One square is designated as your home square.

8.1 Vacuum Rules

Your program is a vacuum that cleans an arbitrary room of grid cells. Make your program an executable that will run from the command line.

Invoke the driver with a list of clients on the command line.

Example: ./vD vacIdiot vacIdiot vacIdiot

A **percept** is something perceived by the agent (the robot). At the start of each turn, you receive a vector of percepts that tell you all the things you can currently perceive. If this were an actual robot instead of a virtual one, the percepts would be provided by physical hardware built into the robot.

All communication with your program is via STDIN and STDOUT. On each turn, your vacuum will receive via STDIN the current set of percepts. On each turn, your vacuum must respond within 1 second with an action. (Don't worry. 1 second is a long time. This is just to let the driver respond properly to failed agents that have dropped into an infinite loop.)

The room is generated at random, on a square-cell pattern, with walls and furnishings filling some of the cells. The vacuum starts at a randomized home location and must finish at that same location.

The driver prompts the vacuum by sending a set of percepts. The percepts are, in order, radar-left, radar-front, radar-right, dirt, and home. Each is a binary quantity, with 1 representing true and 0 representing false. The percepts are space-separated and terminated by newline. For example, "1 0 0 1 0\n".

The vacuum responds by giving a command. The valid commands are: "forward\n": move forward one cell, if possible, else do not move. "left\n": turn left 90 degrees, staying in the same cell. "right\n": turn right 90 degrees, staying in the same cell. "vacuum\n": pick up dirt in the current cell. "off\n": turn off, indicating the task is completed.

For the benefit of human players, the driver will reply with "what?" in case an invalid command is entered. Robotic vacuums should not need this functionality.

For the benefit of robotic players, the driver will echo and ignore any line starting with a "#" mark. This allows the robot to make reports to its programmer to aid in debugging.

The driver creates a visual display of each vacuum's activity, showing the starting random seed, a diagram of the room and vacuum, and a score to date. By using the same random seed, different vacuums can be compared. Score is -100 per dirt remaining, +100 per dirt captured, -1 per command issued, and +100 for ending in the proper cell (direction faced does not matter). Since it is anticipated that all vacuums will seek out all dirt and eliminate it, the score differences will be based on how quickly the task is

accomplished.

Setup:

Download the vD.txt vacuum driver. Rename it if you wish. chmod it to be executable. It is a tcl/expect program.

Download the vA.txt vacuum idiot agent. Rename it if you wish. chmod it to be executable. It is a Perl program.

On the command line, type the following command:

```
./vD vacIdiot vacIdiot vacIdiot vacIdiot
```

This will run the driver with four copies of the “idiot” sample program. (We call it the idiot because it just moves randomly with no planning.)

Create your own program that will behave like the idiot, only lots smarter. Be prepared to submit it in class for competition with agents written by other students.

In the bake off, each vacuum will receive the same starting location and room configuration. All will run until a winner is determined. Total points earned will be compared to determine grades.

8.2 Vacuum Scoring

Success: This task is worth 200 points. For full credit, we expect your vacuum agent to consistently clean the entire room and end up at the designated starting location. We are not so worried about the exact efficiency of your moves, so long as you finish in a reasonable amount of time.

The vacuum that earns the top score will also be the one that completes the task in the least number of moves. We will use that number of moves as our standard. Each vacuum will be given 500 moves initially. Then, if at least one vacuum completed successfully, we will allow all other vacuum double their number of moves before we calculate final scores. For example, if after 500 moves we see that someone completed in 350 moves, we will allow everyone another 200 moves ($350 \times 2 = 700 = 500 + 200$) before we stop the round of competition. If the best score is less than 250 moves, we will stop at 500.

Vacuums that have covered every square possible, and picked up all the dirt possible, and ended at the designated location, all within the moves allotted,

will get full credit.

We will run at least ten rounds of competition, so the vacuums are compared in a variety of randomly generated settings. We will average the results to determine final grades for this task.

Extra Credit: Bonus points may be used to reward agents that are especially efficient. This is calculated by comparing the agent's score with typical scores of all successful agents.

Partial Credit: For agents that fail to cover the entire room, or that fail to end at the designated starting location, points will be awarded in proportion to their final score as reported by the vacuum driver, in comparison to typical scores of all successful agents.

8.3 Vacuum Agent

Following is a sample agent program that is used to illustrate the basics of how the student program could be constructed.

```
#!/usr/bin/perl -w

$moves = 0;
while ( 1 ) {
    chomp ( $line = <STDIN> );
    # print "# $line\n";
    ( $rl, $rf, $rr, $d, $h ) = split ( / /, $line );
    $r = rand(); $moves++;
    if ( $d ) { print "vacuum\n"; next }
    if ( $moves > 1 && $h ) { print "off\n"; last }
    if ( $rf && $rl ) { print "right\n"; next }
    if ( $rf && $rr ) { print "left\n"; next }
    if ( $rl && $rr ) { print "forward\n"; next }
    if ( $rf ) { $dir = "left";
        if ( $r > 0.5 ) { $dir = "right" }
        print "$dir\n"; next }
    if ( $rl ) { $dir = "forward";
        if ( $r > 0.85 ) { $dir = "right" }
        print "$dir\n"; next }
    if ( $rr ) { $dir = "forward";
```

```
    if ( $r > 0.85 ) { $dir = "left" }
    print "$dir\n"; next }
$dir = "forward";
if ( $r > 0.95 ) { $dir = "left" }
if ( $r < 0.05 ) { $dir = "right" }
print "$dir\n";
}
```

Chapter 9

Hunt the Wumpus

Contents

9.1 Wumpus Rules	61
9.2 Wumpus Scoring	62
9.3 Wumpus Agent	62

- Bake Off #1: Thu, 2015-03-05, in class.
- Bake Off #2: Thu, 2015-03-12, in class.
- Bake Off #3: Tue, 2015-03-24, in class.
- Bake Off #4: Thu, 2015-04-16, by 10:00.

Your task in the Wumpus World is to explore a small system of caves, retrieve the gold if possible, avoid death at the hands of the **Wumpus**, and avoid death by falling into one of the pits.

The **Wumpus** is a fictional creature that does not like adventurers, except to eat them.

Your task is to earn points by searching for Gold in a cave system. The system is a four-by-four grid. You begin in the lower left corner (1,1) and you are facing right (East).

You are guaranteed that your initial position and the two squares you can reach are safe from hazards.

You should decide your moves by calculating the probability of various dangers. You initially know that the Wumpus is located in one of the other 13

cells. You also know that the gold is located in one of the 13 cells. You also know that each of the 13 cells has a 20% chance of being a pit.

9.1 Wumpus Rules

Maximize your points, which you gain and lose as follows. Cost: -1 for each action Cost: -10 for shooting your arrow Penalty: -10 for bumping into a wall Penalty: -1000 for death (by pit or Wumpus) Reward: +1000 for getting the gold Reward: +100 for each new room visited / explored Reward: +100 for quitting back at the entrance (1,1) Reward: +100 for killing the Wumpus

Make your program an executable that will run from the command line in a Linux environment.

All communication with your program is via STDIN and STDOUT. On each turn, your hunter will receive via STDIN as set of percepts. On each turn, your hunter must respond within 1 second with an action. (Don't worry. 1 second is a long time unless your program falls into an infinite loop.)

The percepts come as a comma-separated list of five bits of information: (stench,breeze,glitter,bump,scream). Each bit is shown as a letter "y" or a letter "n".

Example: n,n,n,n,n

Stench: "y" means you can smell the Wumpus in an adjacent cell

Breeze: "y" means you can tell there is a pit in an adjacent cell

Glitter: "y" means there is gold in the current cell.

Bump: "y" means you tried to move forward but ran into a wall and did not move.

Scream: "y" means you shot the Wumpus and it is now dead and no longer dangerous. You can still smell it if you are close enough.

In addition to the specified five inputs, the line may contain other inputs that you should ignore. Specifically, it may contain a counter indicating how many moves have been made, but you should not rely on it.

You must reply one of (S,shoot,L,left,R,right,A,forward,G,grab,Q,quit)

Setup:

Download the wD program. chmod it to be executable. It is a tcl/expect program.

For an example, download the wA (idiot) agent. chmod it to be executable. It is a Perl program.

On the command line, type the following command:

```
./wD wA
```

This will run the driver with one copy of the “idiot” sample program. (We call it the idiot because it just moves randomly with no planning.) When one agent is running, the driver displays debug information.

```
./wD wA wA wA wA
```

This will run the driver with four copies of the “idiot” sample program. When several agents are running, the driver does not display debug information.

Create your own program that will behave like the idiot, but much smarter. Be prepared to submit it in class for competition with agents written by other students.

9.2 Wumpus Scoring

Full credit is earned if you successfully visit all the places that are safe, retrieve the gold if possible, shoot the Wumpus if possible, do not bump into any walls, and quit back at the starting location. This should be your goal.

Otherwise, partial credit may be awarded. It will probably be based on your agent’s points in comparison to the best agents. The Wumpus Driver calculates the score, and that is what we will probably use.

9.3 Wumpus Agent

Following is a sample agent program that illustrates the basics of how the student program could be constructed. However, if this is all you do, you will not earn much credit if any.

The sample agent can be downloaded from the course webpage.

```
#!/usr/bin/perl -w

$moves = 0;
while ( 1 ) {
    chomp ( $line = <STDIN> );
    # print "# $line\n";
    ( $stench, $breeze, $glitter, $bump, $scream ) =
        split ( /,/, $line );
    $action = "forward";
    $r = int ( rand ( 4 ) ); $moves++;
    if ( $r == 0 ) { $action = "left" }
    if ( $r == 1 ) { $action = "right" }
    print "$action\n";
}
```


Chapter 10

Number Recognition

Contents

10.1 Preparation	65
10.2 Background	65
10.3 Phonemes	66
10.4 Vocabulary	68
10.5 Over Training	68
10.6 Numbers Scoring	69
10.7 Evaluation	69
10.8 Testing Your Program	70

- Bake Off #1: Thu, 2015-04-02, in class.
- Bake Off #2: Thu, 2015-04-09, in class.
- Bake Off #3: Tue, 2015-04-14, in class.
- Bake Off #4: Thu, 2015-04-16, by 10:00.

Your task is to write a program that converts **phonemes** into **numbers**. The phonemes are presented as a time-aligned transcription where each phoneme has a starting time, an ending time, and a symbolic representation. The resulting numbers are to be expressed in written characters as whole individual words.

10.1 Preparation

On the IS2 machine, set up a test bed. Create a directory where you will do your testing. You can follow these steps:

```
cd yourtestdirectory
ln -s ~dc/cs440/numbers/data data
cp data/nD driver
cp data/nIdiot sample
./driver ./sample data/0/*
```

`ln -s ~dc/cs440/numbers/data data` gives you ready access to the files you will use for testing.

`cp data/nD driver` gives you a copy of the test driver. You can use this just as it is.

`cp data/nIdiot sample` gives you a copy of a very simple-minded recognition program. What it really shows you is how to bring in the inputs and put out the outputs.

`data/0/*` gives you one tenth of all the data files. The driver program will run your recognition program repeatedly, each time with a different set of inputs. If your program gives the right answer, you get points. If not, it tells you what the inputs were and the expected and actual outputs. Then you fix your program and test again.

You can write your program anywhere. You can upload it for testing. The testing will happen on the IS2 machine.

10.2 Background

CSLU is the Center for Spoken Language Understanding. I am familiar with it because it was the home to my PhD program back in the 1990s.

We will use as our standard the CSLU Numbers corpus. A sample of this corpus is available for free download from the CSLU Corpus website. At this writing, the homepage for CSLU corpora is:

<http://www.cslu.ogi.edu/corpora/corpCurrent.html>

Find the Numbers Corpus, probably version 1.3.

Download the sample file, probably called `numbers_sample.zip` and store it somewhere so you can use it for the remainder of this project.

Open the file and review its contents. You should find a folder (directory) named “labels” and another folder named “trans”. The labels directory contains the time-aligned phonetic labels in files each with a `.phn` extension. The trans directory contains the orthographic (normal writing) transcriptions in files each with a `.txt` extension.

All files are in plain ASCII text.

Display: Your instructor may ask you to display a matching pair of files, one `.phn` and one `.txt`, to show that you understand the file organization and to verify that you can work with the files.

10.3 Phonemes

Phonemes are sounds from which utterances are built up. Commonly phonemes can be divided into two categories: vowels and consonants.

The word “cat” generally is spoken using the three phonemes “k”, “a”, “t”.

The word “Kate” generally is spoken using the four phonemes “k”, “eh”, “ee”, “t”.

IPA, Worldbet, and OGIbet English Broad Phonetic Labels

There are around 50 different phonemes in English. You can find them described on the chart on the next page.

IPA, Worldbet, and OGIbet English Broad Phonetic Labels

Center for Spoken Language Understanding – Oregon Graduate Institute of Science & Technology

IPA	Worldbet	OGIbet	Example	Category
i:	i:	iy	be <u>e</u> t	Front Vowels
I	I	ih	bi <u>i</u> t	
ε	E	eh	be <u>e</u> t	
æ	@	ae	ba <u>a</u> t	
ɪ	I_x	ix	rose <u>s</u>	Central Vowels (British)
ʊ	u_x	ux	sui <u>t</u>	
ə	&	ax	ab <u>o</u> ve	
ə	&0		to go	
ɒ	5		po <u>t</u>	
u	u	uw	bo <u>o</u> t	Back Vowels
ʊ	U	uh	bo <u>o</u> k	
ʌ	^	ah	ab <u>o</u> ve	
ɔ	>	ao	ca <u>u</u> ght	
ɑ	A	aa	fa <u>t</u> her	
ɝ	3r	er	bi <u>r</u> d	Retroflexes
ɞ	&r	axr	bu <u>t</u> ter	
ei	ei	ey	ba <u>y</u>	Diphthongs (British) (British) (British)
aI	aI	ay	by <u>e</u>	
ɔi	>i	oy	bo <u>y</u>	
iʊ	iU		fe <u>w</u>	
aʊ	aU	aw	abo <u>u</u> t	
oʊ	oU	ow	bo <u>a</u> t	
iə	i&		he <u>r</u> e	
eə	e&		the <u>r</u> e	
uə	u&		po <u>o</u> r	
p ^h	ph	p	pa <u>n</u>	Voiceless Plosives
t ^h	th	t	ta <u>n</u>	
k ^h	kh	k	ca <u>n</u>	
b	b	b	ba <u>n</u>	Voiced Plosives
d	d	d	da <u>n</u>	
g	g	g	ga <u>n</u> der	
m	m	m	me <u> </u>	Nasals
n	n	n	kn <u>ee</u>	
ŋ	N	ng	si <u>ng</u>	
r _l	th_ (dx	wri <u>ter</u>	Flaps
r _d	d_ (dx	ri <u>der</u>	
f	f	f	fi <u>n</u> e	Voiceless Fricatives
θ	T	th	thi <u>gh</u>	
s	s	s	si <u>gn</u>	
ʃ	S	sh	assu <u>r</u> e	
h	h	hh	ho <u>p</u> e	
v	v	v	vi <u>n</u> e	Voiced Fricatives
ð	D	dh	thi <u> </u>	
z	z	z	resi <u>gn</u>	
ʒ	Z	zh	azu <u>r</u> e	
tʃ	tS	ch	ch <u>ur</u> ch	Affricates
dʒ	dZ	jh	ju <u>dg</u> e	
l	l	l	le <u>n</u> t	Glides (approximants)
ɹ	9r	r	re <u>n</u> t	
j	j	y	ye <u>s</u>	
w	w	w	w <u>e</u> nt	
m	m=	em	bot <u>to</u> m	Syllabics
n	n=	en	but <u>to</u> n	
ŋ	N=	eng		
l	l=	el	bot <u>tl</u> e	

IPA	Worldbet	OGIbet	Example	Category
	pc	pcl	_pa <u>n</u>	Voiceless
	tc	tc1	_ta <u>n</u>	Plosive
	kc	kcl	_ca <u>n</u>	Closures
	bc	bcl	_ba <u>n</u>	Voiced
	dc	dcl	_da <u>n</u>	Plosive
	gc	gcl	_ga <u>n</u> der	Closures
	tSc	chcl	_ch <u>ur</u> ch	Affricate
	dZc	jhcl	_ju <u>dg</u> e	Closures
	+	.epi	epi <u>n</u> thetic closure	

IPA	Worldbet	OGIbet	Type of Diacritic
t ^h	_h	-h	aspirated
	_x		centralized
t̥ d̥	_l		dental
	_ (flapped (consonant)
	_F		fricated stop
	_?*	q	glottal onset
ʔ	_?	-q	glottalized
d ^l	_l		lateral release
i:	_:	-el	lengthened
d ⁿ	_n		nasal release
ẽ	_~	-n	nasalized
	_NL	.nitl	not in the language
t ^j	_j		palatalized
ɝ	_r	-r	retroflexion
ɚ	_i		less rounded
ɜ	_w		more rounded
ɹ	_=		syllabicity
ɣ	_v		voiced
n̥ d̥	_0		voiceless
	_*	-	waveform cut off

<i>Worldbet, as modified at OGI</i>			
	_fp	-fp	filled pause
	_ln	-ln	line noise corruption
	_bn		background noise

Worldbet	OGIbet	Non Speech Sound Item
.bn	.bn	background noise
.br	.br	breath noise
.cough	.cough	cough
.ct	.ct	clear throat
.laugh	.laugh	laugh
.ln	.ln	lin noise
.ls	.ls	lip smack
.ns	.ns	human, not speech
.sneeze	.sneeze	sneeze
.tc	.tc	tongue click

<i>Worldbet, as modified at OGI</i>		
.beep	.beep	beep
.burp	.burp	burp
.fp	.fp	filled pause
.pau	.pau	pause or silence
.sniff	.sniff	sniff
.uu	.unk	unintelligible speech
.vs	.vs	squeak, voice crack
.glot	glot	glottalization

10.4 Vocabulary

The full CSLU Numbers corpus consists of about 23,900 utterances, each with a phonetic transcription and an orthographic transcription. The transcriptions were prepared by trained human transcribers.

The sample corpus consists of 283 transcribed utterances, or about one percent of the full corpus.

The following (64 or so) words will be counted in the scoring.

a and double eight eighteen eighteenth eighth eightieth eighty eleven eleventh fifteen fifteenth fifth fiftieth fifty first five fortieth forty four fourteen fourteenth fourth half hundred hundredth nine nineteen nineteenth ninetieth ninety ninth oh one second seven seventeen seventeenth seventh seventieth seventy six sixteen sixteenth sixth sixtieth sixty ten tenth third thirteen thirteenth thirtieth thirty thousand three triple twelfth twelve twentieth twenty two zero

All other words and fragments of words will be ignored.

However, you may need to account for the fact that some non-scored words will occur in the utterances that you are processing.

10.5 Over Training

Your program is performing a recognition task. There are two ways to approach such a task. One is by strict memorization. You check to see which of the 283 cases you are viewing and you report the correct results for that case.

The other is by a careful combination of memorization (**basis** cases) and rules (**induction**). Rather than memorizing the 283 sample results, you may memorize the 46 words that occur in the sample results. And you may create suitable rules to combine the memorized results to make additional, non-memorized results.

A brief example may help. When we calculate $4+3$, and we are very small, we may hold up four fingers on one hand and three on the other hand. Then we may count all the fingers to arrive at a total of seven. In this example, the meaning of 4 was memorized, the meaning of 3 was memorized, and the process of combining them was memorized. 4 and 3 are part of the basis.

The process is part of the induction.

Later in life, we may actually memorize our “addition facts” or “addition tables” and come to know that $4+3=7$ without ever counting it out. We can tackle a problem like $639+15$ by breaking it up into smaller problems: $9+5=14$; write down the 4 and carry the 1. $3+1+1$ is 5; write it down and carry zero. 6 comes down for a total of 654. In this process, the single-digit additions were memorized and the rules for carry and column order provide the inductive step to we can add numbers of any size.

Because this same situation occurs in recognizing numbers, we must be careful to keep our basis set as small as possible, within reason. It is good to divide your corpus (the 283 utterances) into parts. One part can be used for development. You can look at those utterances in great detail and analyze them to understand your task. Another part can be used for testing. You use those utterances to see how well your program performs. The great risk is that you will train your program to recognize your 283 utterances very well, but you will fail to recognize the other 23,600 utterances. That is called **over training**. Use your resources carefully.

10.6 Numbers Scoring

On a 200-point basis, scoring is as follows.

For a accuracy rates from 0 to 40%, the grade is (accuracy * 5 / 4). The grade range is 0 to 50 points.

For a accuracy rates from 40 to 70%, the grade is (50 points plus (accuracy above 40 * 5 / 3)). The grade range is 50 to 100 points.

For a accuracy rates from 70 to 90%, the grade is (100 points plus (accuracy above 70 * 5 / 2)). The grade range is 100 to 150 points.

For a accuracy rates above 90%, the grade is (150 points plus (accuracy above 90 * 10)). The grade range is 150 to 250 points. (Above 95% accuracy is extra credit.)

10.7 Evaluation

I intend to give you more than the 283 utterances that are in the sample set. If so, you must not disclose them beyond this class. That is part of the

license agreement I signed in order to get the corpus. You can study the additional utterances to improve your performance.

When I test your program, I will report the number of utterances it correctly recognized, meaning that the entire utterance was exactly right in every way. I will provide you with my testing script, but not with the data files that I am using until afterward.

The input files (.phn) are given to you in their original form, redirected to you as STDIN standard input, with results captured from you as STDOUT standard out. Your file should present one word per line, interspersed at your discretion with comment lines that I will ignore. Comment lines will be identified by a “#” as the first character of the line.

Each corpus transcription file (.txt) will be “normalized” by being stripped of all words that start with dot (such as .pau), and all characters that are enclosed in angle brackets (such as <bn>). Letters will be converted to lower case. Runs of white space will be converted to a single space. The 28 characters a-z, apostrophe, and space, will be kept. All other characters will be deleted.

Most but not all human transcriptions are correct. A few are wrong, so it is impossible to get a perfect score. Also some transcriptions include word fragments. But try to get the best score you can. As I verify incorrect transcriptions, I do try to remove them from the corpus against which you are tested.

10.8 Testing Your Program

Here is a sample program you can key in and test.

```
#!/usr/bin/perl --
while ( $in = <STDIN> ) { print "# got input: $in" }
print "one two three four five\n" ;# your answer
```

You can test it by following these steps.

- (1) Create a directory for this project in your account on the is2 machine.
- (2) Inside that directory, create these symbolic links:

```
ln -s ~dc/cs440/numbers/data
```

```
ln -s ~dc/cs440/numbers/nDriver
```

(3) Put your program in your directory. We will assume you have called it "Don0".

(4) Test your program by running the following command line.

```
./nDriver ./Don0 data/sample/NU-1*
```

(5) You should see two program executions, with details of your program's operation.

The summary tells your accuracy, (number correct / number tested), and your score, $(-1000 * \log_2(\text{error rate}))$, max 9999.

Every time you cut your error rate in half, your score goes up by 1000.

An accuracy of 0.00 (none right) results in a score of 0.

An accuracy of 0.50 results in a score of 1000.

An accuracy of 0.75 results in a score of 2000.

An accuracy of 0.875 results in a score of 3000.

The maximum score is 9999.

Driving down your error rate really drives up your score.

(6) Now improve your program to print the correct answer each time.

(7) You can submit your program to the instructor by copying it as follows:

```
cp Don0 ~dc/inbox/
```


Chapter 11

Phonetics

This chapter is a collection of thoughts that might someday prove useful.

Some utterances are phonetically identical but differ in meaning.

“He earned a C” versus “He yearned to see.”

“Fork handles” versus “Four candles.”

<http://en.wikipedia.org/wiki/Mondegreen>

Chapter 12

Exam Topics

Note: This semester I do not plan to offer a final exam of the type mentioned here, but I want to keep this material in the study guide for future use.

On the final exam you may be asked to discuss important topics from the textbook or our class discussions. You will write from memory about the subject, telling what it is, why it is important, and other comments you may wish to share. Students typically write for two to three minutes on each topic.

Following is a preliminary list of topics based on prior semesters readings and discussions.

In addition to this list, consider the learning objectives mentioned in chapter [A](#) (page 80).

Exam Question 34 (p.92): Discuss: Artificial Intelligence

Hint: Sec 1.0

Exam Question 35 (p.92): Discuss: Turing test

Hint: Sec 1.1: If you cannot distinguish its performance from that of a human, then it we can call it intelligent.

Exam Question 36 (p.92): Discuss: automated reasoning

Hint: Sec 1.1: Use knowledge base to derive new conclusions.

Exam Question 37 (p.92): Discuss: machine learning

Hint: Sec 1.1: Adapt to new circumstances. Detect and extrapolate patterns.

Exam Question 38 (p.92): Discuss: total Turing test

Hint: Sec 1.1: Includes video and ability to pass physical objects through the hatch to test perceptual abilities.

Exam Question 39 (p.92): Discuss: agent

Hint: Sec 1.1: A program that is autonomous, acts, perceives, persists, adapts to change, and pursues the goals it was given.

Exam Question 40 (p.92): Discuss: dualism

Hint: Sec 1.2: Part of us exists outside of nature. We have a spirit. Mysticism.

Exam Question 41 (p.92): Discuss: materialism

Hint: Sec 1.2: None of us exists outside of nature. We are just meat.

Exam Question 42 (p.92): Discuss: logical positivism

Hint: Sec 1.2: All knowledge can be expressed as logic sentences.

Exam Question 43 (p.92): Discuss: algorithm

Hint: Sec 1.2: A set of steps that a program follows.

Exam Question 44 (p.92): Discuss: Godel's incompleteness theorem

Hint: Sec 1.2: There are true statements that cannot be proved by any algorithm.

Exam Question 45 (p.92): Discuss: intractability

Hint: Sec 1.2: Time to solve grows exponentially with size of input. (Cannot be pulled apart into smaller problems.)

Exam Question 46 (p.92): Discuss: NP-completeness

Hint: Sec 1.2: Probably intractable.

Exam Question 47 (p.92): Discuss: machine evolution

Hint: Sec 1.3: Genetic algorithms. Code mutation and survival of the fittest.

Exam Question 48 (p.92): Discuss: genetic algorithms

Hint: Sec 1.3: Machine evolution. Code mutation and survival of the fittest.

Exam Question 49 (p.92): Discuss: expert systems

Hint: Sec 1.3: Rules are derived from interviewing experts. Rules reflect uncertainty.

Exam Question 50 (p.92): Discuss: frames

Hint: Sec 1.3: Structures into which knowledge is arranged.

Exam Question 51 (p.92): Discuss: rational agent

Hint: Sec 2.2: Agent always does the right thing.

Exam Question 52 (p.92): Discuss: autonomous agent

Hint: Sec 2.2: Learns beyond what it was originally told.

Exam Question 53 (p.92): Discuss: simple reflex agent

Hint: Sec 2.4 p46: No memory. Responds immediately to what is around it.

Exam Question 54 (p.92): Discuss: goal-based agent

Hint: Sec 2.4 p49: You give it a specific goal. It uses logic to get there.

Exam Question 55 (p.92): Discuss: utility-based agent

Hint: Sec 2.4 p51: The goal is to maximize utility.

Exam Question 56 (p.93): Discuss: accessible environment

Hint: Sec 2.4: Agent can know everything about the environment. Nothing is hidden.

Exam Question 57 (p.93): Discuss: deterministic environment

Hint: Sec 2.4: Environment is totally predictable. There is no randomness in how it responds to the actions of the agent.

Exam Question 58 (p.93): Discuss: episodic environment

Hint: Sec 2.4: Parts picker.

Exam Question 59 (p.93): Discuss: static vs dynamic environment

Hint: Sec 2.4

Exam Question 60 (p.93): Discuss: discrete vs continuous environment

Hint: Sec 2.4

Exam Question 61 (p.93): Discuss: search

Hint: Sec 3.0: No single action will do.

Exam Question 62 (p.93): Discuss: path cost

Hint: Sec 3.1 p62: weighted cost of actions, total cost of path.

Exam Question 63 (p.93): Discuss: breadth-first search

Hint: Sec 3.4 p73: Uses a queue. Uses lots of memory.

Exam Question 64 (p.93): Discuss: uniform-cost search

Hint: Sec 3.4 p75: Uses priority queue. Compares cost, not depth. Finds optimal solution if all costs gt 0. Uses lots of memory.

Exam Question 65 (p.93): Discuss: depth-first search

Hint: Sec 3.4 p75: expands deepest first. may search forever. Gentle on

memory.

Exam Question 66 (p.93): Discuss: depth-limited search

Hint: Sec 3.4 p77: Finds optimal solution if it exists within the limit. Gentle on memory.

Exam Question 67 (p.93): Discuss: iterated deepening search

Hint: Sec 3.4 p78: Finds optimal solution. Gentle on memory. Takes more time than BFS but not much.

Exam Question 68 (p.93): Discuss: bidirectional search

Hint: Sec 3.4 p79: Search from start and goal. Stop when they touch.

Exam Question 69 (p.93): Discuss: heuristics

Hint: Sec 4.1 p95: estimate, esp cost of cheapest path to the goal.

Exam Question 70 (p.93): Discuss: best-first search

Hint: Sec 4.x p129: Expands the minimum-cost node next.

Exam Question 71 (p.93): Discuss: greedy search

Hint: Sec 4.1 p95: greedy search. BFS sort by heuristic remaining cost only.

Exam Question 72 (p.93): Discuss: A* search

Hint: Sec 4.1 p97: A* search. BFS sort by current cost plus heuristic remaining cost.

Exam Question 73 (p.93): Discuss: admissible heuristic

Hint: Sec 4.1 p97: A* heuristic that never over-estimates remaining cost.

Exam Question 74 (p.93): Discuss: knowledge representation

Hint: Sec 10.x

Exam Question 75 (p.93): Discuss: inference (sound, complete)

Hint: Sec 7.1

Exam Question 76 (p.93): Discuss: propositional logic

Hint: Sec 7.x

Exam Question 77 (p.93): Discuss: first-order logic

Hint: Sec 8.x

Exam Question 78 (p.93): Discuss: atomic sentence

Hint: Sec 8.x

Exam Question 79 (p.93): Discuss: predicate

Hint: Sec 8.x

Exam Question 80 (p.94): Discuss: quantified sentence

Hint: Sec 8.x

Exam Question 81 (p.94): Discuss: situational calculus

Hint: Sec 8.x

Exam Question 82 (p.94): Discuss: diagnostic rules

Hint: Sec 8.x

Exam Question 83 (p.94): Discuss: causal rules

Hint: Sec 8.x

Exam Question 84 (p.94): Discuss: unification

Hint: Sec 9.x

Exam Question 85 (p.94): Discuss: Modus Ponens

Hint: Sec 9.x

Exam Question 86 (p.94): Discuss: Horn form

Hint: Sec 9.x

Exam Question 87 (p.94): Discuss: resolution

Hint: Sec 9.x

Exam Question 88 (p.94): Discuss: conjunctive normal form

Hint: Sec 9.x

Exam Question 89 (p.94): Discuss: implicative normal form

Hint: Sec 9.x

Exam Question 90 (p.94): Discuss: conditional plans

Hint: Sec 11.x

Exam Question 91 (p.94): Discuss: execution monitoring

Hint: Sec 11.x

Exam Question 92 (p.94): Discuss: action monitoring

Hint: Sec 11.x

Exam Question 93 (p.94): Discuss: replanning agent

Hint: Sec 12.x

Exam Question 94 (p.94): Discuss: prior probabilities

Hint: Sec 14.x

Exam Question 95 (p.94): Discuss: conditional probabilities

Hint: Sec 14.x

Exam Question 96 (p.94): Discuss: joint probability distribution

Hint: Sec 14.x

Exam Question 97 (p.94): Discuss: Bayes' rule

Hint: Sec 14.x

Exam Question 98 (p.94): Discuss: conditional independence

Hint: Sec 14.x

Exam Question 99 (p.94): Discuss: Bayesian updating

Hint: Sec 14.x

Exam Question 100 (p.94): Discuss: belief networks

Hint: Sec 15.x

Exam Question 101 (p.94): Discuss: stochastic simulation

Hint: Sec 15.x

Exam Question 102 (p.94): Discuss: truth-functional system

Hint: Sec 15.x

Exam Question 103 (p.94): Discuss: performance element

Hint: Sec 18.x

Exam Question 104 (p.95): Discuss: learning element

Hint: Sec 18.x

Exam Question 105 (p.95): Discuss: inductive learning

Hint: Sec 18.x

Exam Question 106 (p.95): Discuss: neural network

Hint: Sec 19.x

Exam Question 107 (p.95): Discuss: perceptron

Hint: Sec 19.x

Exam Question 108 (p.95): Discuss: linearly separable function

Hint: Sec 19.x

Exam Question 109 (p.95): Discuss: feed-forward network

Hint: Sec 19.x

Exam Question 110 (p.95): Discuss: back-propagation

Hint: Sec 19.x

Exam Question 111 (p.95): Discuss: Bayesian learning

Hint: Sec 19.x

Exam Question 112 (p.95): Discuss: multi-layer feed-forward network

Hint: Sec 19.x

Exam Question 113 (p.95): Discuss: speech act

Hint: Sec 22.1 p790: the action of creating an information-bearing message.

Exam Question 114 (p.95): Discuss: phrase-structure grammar

Hint: Sec 22.x

Exam Question 115 (p.95): Discuss: context-free grammar

Hint: Sec 22.x

Exam Question 116 (p.95): Discuss: encoded message

Hint: Sec 22.x

Exam Question 117 (p.95): Discuss: situated language

Hint: Sec 22.x

Exam Question 118 (p.95): Discuss: augmented grammar

Hint: Sec 22.x

Exam Question 119 (p.95): Discuss: pragmatic interpretation

Hint: Sec 22.x

Exam Question 120 (p.95): Discuss: disambiguation

Hint: Sec 22.x

Exam Question 121 (p.95): Discuss: anytime algorithm

Hint: Sec 27.1

Exam Question 122 (p.95): Discuss: bounded optimality

Hint: Sec 27.2

Exam Question 123 (p.95): Discuss: prisoner's dilemma

Hint: Sec 27.2

Appendix A

CC2001: Intelligent Systems

The following eight pages are taken from CC2001: Computing Curricula 2001, Computer Science. It represents the 2001 view of computer science by the curriculum committees of **ACM**: the Association for Computing Machinery, and **IEEE-CS**: the Computer Society of the Institute for Electrical and Electronic Engineers.

In this class, we will be mostly interested in the following learning objectives.

IS1: 1, 2, 3, 5.

IS2: 1, 2, 5.

IS3: 1, 2, 3, 4.

IS6: 1, 2, 4, 6.

IS7: 3.

These would be good topics to investigate in your weekly readings.



Computing Curricula 2001

Computer Science

— Final Report —
(December 15, 2001)

The Joint Task Force on Computing Curricula
IEEE Computer Society
Association for Computing Machinery

This material is based upon work supported by the
National Science Foundation under Grant No. 0003263

Figure 5-1. Computer science body of knowledge with core topics underlined

<p>DS. Discrete Structures (43 core hours) <u>DS1. Functions, relations, and sets</u> (6) <u>DS2. Basic logic</u> (10) DS3. Proof techniques (12) <u>DS4. Basics of counting</u> (5) <u>DS5. Graphs and trees</u> (4) <u>DS6. Discrete probability</u> (6)</p> <p>PF. Programming Fundamentals (38 core hours) <u>PF1. Fundamental programming constructs</u> (9) <u>PF2. Algorithms and problem-solving</u> (6) <u>PF3. Fundamental data structures</u> (14) <u>PF4. Recursion</u> (5) <u>PF5. Event-driven programming</u> (4)</p> <p>AL. Algorithms and Complexity (31 core hours) <u>AL1. Basic algorithmic analysis</u> (4) <u>AL2. Algorithmic strategies</u> (6) <u>AL3. Fundamental computing algorithms</u> (12) <u>AL4. Distributed algorithms</u> (3) <u>AL5. Basic computability</u> (6) AL6. The complexity classes P and NP AL7. Automata theory AL8. Advanced algorithmic analysis AL9. Cryptographic algorithms AL10. Geometric algorithms AL11. Parallel algorithms</p> <p>AR. Architecture and Organization (36 core hours) <u>AR1. Digital logic and digital systems</u> (6) <u>AR2. Machine level representation of data</u> (3) <u>AR3. Assembly level machine organization</u> (9) <u>AR4. Memory system organization and architecture</u> (5) <u>AR5. Interfacing and communication</u> (3) <u>AR6. Functional organization</u> (7) <u>AR7. Multiprocessing and alternative architectures</u> (3) AR8. Performance enhancements AR9. Architecture for networks and distributed systems</p> <p>OS. Operating Systems (18 core hours) <u>OS1. Overview of operating systems</u> (2) <u>OS2. Operating system principles</u> (2) <u>OS3. Concurrency</u> (6) <u>OS4. Scheduling and dispatch</u> (3) <u>OS5. Memory management</u> (5) OS6. Device management OS7. Security and protection OS8. File systems OS9. Real-time and embedded systems OS10. Fault tolerance OS11. System performance evaluation OS12. Scripting</p> <p>NC. Net-Centric Computing (15 core hours) <u>NC1. Introduction to net-centric computing</u> (2) <u>NC2. Communication and networking</u> (7) <u>NC3. Network security</u> (3) <u>NC4. The web as an example of client-server computing</u> (3) NC5. Building web applications NC6. Network management NC7. Compression and decompression NC8. Multimedia data technologies NC9. Wireless and mobile computing</p> <p>PL. Programming Languages (21 core hours) <u>PL1. Overview of programming languages</u> (2) <u>PL2. Virtual machines</u> (1) <u>PL3. Introduction to language translation</u> (2) <u>PL4. Declarations and types</u> (3) <u>PL5. Abstraction mechanisms</u> (3) <u>PL6. Object-oriented programming</u> (10) PL7. Functional programming PL8. Language translation systems PL9. Type systems PL10. Programming language semantics PL11. Programming language design</p>	<p>HC. Human-Computer Interaction (8 core hours) <u>HC1. Foundations of human-computer interaction</u> (6) <u>HC2. Building a simple graphical user interface</u> (2) HC3. Human-centered software evaluation HC4. Human-centered software development HC5. Graphical user-interface design HC6. Graphical user-interface programming HC7. HCI aspects of multimedia systems HC8. HCI aspects of collaboration and communication</p> <p>GV. Graphics and Visual Computing (3 core hours) <u>GV1. Fundamental techniques in graphics</u> (2) <u>GV2. Graphic systems</u> (1) GV3. Graphic communication GV4. Geometric modeling GV5. Basic rendering GV6. Advanced rendering GV7. Advanced techniques GV8. Computer animation GV9. Visualization GV10. Virtual reality GV11. Computer vision</p> <p>IS. Intelligent Systems (10 core hours) <u>IS1. Fundamental issues in intelligent systems</u> (1) <u>IS2. Search and constraint satisfaction</u> (5) <u>IS3. Knowledge representation and reasoning</u> (4) IS4. Advanced search IS5. Advanced knowledge representation and reasoning IS6. Agents IS7. Natural language processing IS8. Machine learning and neural networks IS9. AI planning systems IS10. Robotics</p> <p>IM. Information Management (10 core hours) <u>IM1. Information models and systems</u> (3) <u>IM2. Database systems</u> (3) <u>IM3. Data modeling</u> (4) IM4. Relational databases IM5. Database query languages IM6. Relational database design IM7. Transaction processing IM8. Distributed databases IM9. Physical database design IM10. Data mining IM11. Information storage and retrieval IM12. Hypertext and hypermedia IM13. Multimedia information and systems IM14. Digital libraries</p> <p>SP. Social and Professional Issues (16 core hours) <u>SP1. History of computing</u> (1) <u>SP2. Social context of computing</u> (3) <u>SP3. Methods and tools of analysis</u> (2) <u>SP4. Professional and ethical responsibilities</u> (3) <u>SP5. Risks and liabilities of computer-based systems</u> (2) <u>SP6. Intellectual property</u> (3) <u>SP7. Privacy and civil liberties</u> (2) SP8. Computer crime SP9. Economic issues in computing SP10. Philosophical frameworks</p> <p>SE. Software Engineering (31 core hours) <u>SE1. Software design</u> (8) <u>SE2. Using APIs</u> (5) <u>SE3. Software tools and environments</u> (3) <u>SE4. Software processes</u> (2) <u>SE5. Software requirements and specifications</u> (4) <u>SE6. Software validation</u> (3) <u>SE7. Software evolution</u> (3) <u>SE8. Software project management</u> (3) SE9. Component-based computing SE10. Formal methods SE11. Software reliability SE12. Specialized systems development</p> <p>CN. Computational Science (no core hours) CN1. Numerical analysis CN2. Operations research CN3. Modeling and simulation CN4. High-performance computing</p>
---	---

Note: The numbers in parentheses represent the minimum number of hours required to cover this material in a lecture format. It is always appropriate to include more.

Intelligent Systems (IS)

- IS1. Fundamental issues in intelligent systems [core]**
- IS2. Search and constraint satisfaction [core]**
- IS3. Knowledge representation and reasoning [core]**
- IS4. Advanced search [elective]**
- IS5. Advanced knowledge representation and reasoning [elective]**
- IS6. Agents [elective]**
- IS7. Natural language processing [elective]**
- IS8. Machine learning and neural networks [elective]**
- IS9. AI planning systems [elective]**
- IS10. Robotics [elective]**

The field of artificial intelligence (AI) is concerned with the design and analysis of autonomous agents. These are software systems and/or physical machines, with sensors and actuators, embodied for example within a robot or an autonomous spacecraft. An intelligent system has to perceive its environment, to act rationally towards its assigned tasks, to interact with other agents and with human beings.

These capabilities are covered by topics such as computer vision, planning and acting, robotics, multiagents systems, speech recognition, and natural language understanding. They rely on a broad set of general and specialized knowledge representations and reasoning mechanisms, on problem solving and search algorithms, and on machine learning techniques.

Furthermore, artificial intelligence provides a set of tools for solving problems that are difficult or impractical to solve with other methods. These include heuristic search and planning algorithms, formalisms for knowledge representation and reasoning, machine learning techniques, and methods applicable to sensing and action problems such as speech and language understanding, computer vision, and robotics, among others. The student needs to be able to determine when an AI approach is appropriate for a given problem, and to be able to select and implement a suitable AI method.

IS1. Fundamental issues in intelligent systems [core]

Minimum core coverage time: 1 hour

Topics:

- History of artificial intelligence
- Philosophical questions
 - The Turing test
 - Searle’s “Chinese Room” thought experiment
 - Ethical issues in AI
- Fundamental definitions
 - Optimal vs. human-like reasoning
 - Optimal vs. human-like behavior
- Philosophical questions
- Modeling the world
- The role of heuristics

Learning objectives:

1. Describe the Turing test and the “Chinese Room” thought experiment.
2. Differentiate the concepts of optimal reasoning and human-like reasoning.
3. Differentiate the concepts of optimal behavior and human-like behavior.

4. List examples of intelligent systems that depend on models of the world.
5. Describe the role of heuristics and the need for tradeoffs between optimality and efficiency.

IS2. Search and constraint satisfaction [core]

Minimum core coverage time: 5 hours

Topics:

Problem spaces
Brute-force search (breadth-first, depth-first, depth-first with iterative deepening)
Best-first search (generic best-first, Dijkstra's algorithm, A*, admissibility of A*)
Two-player games (minimax search, alpha-beta pruning)
Constraint satisfaction (backtracking and local search methods)

Learning objectives:

1. Formulate an efficient problem space for a problem expressed in English by expressing that problem space in terms of states, operators, an initial state, and a description of a goal state.
2. Describe the problem of combinatorial explosion and its consequences.
3. Select an appropriate brute-force search algorithm for a problem, implement it, and characterize its time and space complexities.
4. Select an appropriate heuristic search algorithm for a problem and implement it by designing the necessary heuristic evaluation function.
5. Describe under what conditions heuristic algorithms guarantee optimal solution.
6. Implement minimax search with alpha-beta pruning for some two-player game.
7. Formulate a problem specified in English as a constraint-satisfaction problem and implement it using a chronological backtracking algorithm.

IS3. Knowledge representation and reasoning [core]

Minimum core coverage time: 4 hours

Topics:

Review of propositional and predicate logic
Resolution and theorem proving
Nonmonotonic inference
Probabilistic reasoning
Bayes theorem

Learning objectives:

1. Explain the operation of the resolution technique for theorem proving.
2. Explain the distinction between monotonic and nonmonotonic inference.
3. Discuss the advantages and shortcomings of probabilistic reasoning.
4. Apply Bayes theorem to determine conditional probabilities.

IS4. Advanced search [elective]

Topics:

- Genetic algorithms
- Simulated annealing
- Local search

Learning objectives:

1. Explain what genetic algorithms are and contrast their effectiveness with the classic problem-solving and search techniques.
2. Explain how simulated annealing can be used to reduce search complexity and contrast its operation with classic search techniques.
3. Apply local search techniques to a classic domain.

IS5. Advanced knowledge representation and reasoning [elective]

Topics:

- Structured representation
 - Frames and objects
 - Description logics
 - Inheritance systems
- Nonmonotonic reasoning
 - Nonclassical logics
 - Default reasoning
 - Belief revision
 - Preference logics
 - Integration of knowledge sources
 - Aggregation of conflicting belief
- Reasoning on action and change
 - Situation calculus
 - Event calculus
 - Ramification problems
- Temporal and spatial reasoning
- Uncertainty
 - Probabilistic reasoning
 - Bayesian nets
 - Fuzzy sets and possibility theory
 - Decision theory
- Knowledge representation for diagnosis, qualitative representation

Learning objectives:

1. Compare and contrast the most common models used for structured knowledge representation, highlighting their strengths and weaknesses.
2. Characterize the components of nonmonotonic reasoning and its usefulness as a representational mechanisms for belief systems.
3. Apply situation and event calculus to problems of action and change.
4. Articulate the distinction between temporal and spatial reasoning, explaining how they interrelate.
5. Describe and contrast the basic techniques for representing uncertainty.
6. Describe and contrast the basic techniques for diagnosis and qualitative representation.

IS6. Agents [elective]

Topics:

- Definition of agents
- Successful applications and state-of-the-art agent-based systems
- Agent architectures
 - Simple reactive agents
 - Reactive planners
 - Layered architectures
 - Example architectures and applications
- Agent theory
 - Commitments
 - Intentions
 - Decision-theoretic agents
 - Markov decision processes (MDP)
- Software agents, personal assistants, and information access
 - Collaborative agents
 - Information-gathering agents
- Believable agents (synthetic characters, modeling emotions in agents)
- Learning agents
- Multi-agent systems
 - Economically inspired multi-agent systems
 - Collaborating agents
 - Agent teams
 - Agent modeling
 - Multi-agent learning
- Introduction to robotic agents
- Mobile agents

Learning objectives:

1. Explain how an agent differs from other categories of intelligent systems.
2. Characterize and contrast the standard agent architectures.
3. Describe the applications of agent theory, to domains such as software agents, personal assistants, and believable agents.
4. Describe the distinction between agents that learn and those that don't.
5. Demonstrate using appropriate examples how multi-agent systems support agent interaction.
6. Describe and contrast robotic and mobile agents.

IS7. Natural language processing [elective]

Topics:

- Deterministic and stochastic grammars
- Parsing algorithms
- Corpus-based methods
- Information retrieval
- Language translation
- Speech recognition

Learning objectives:

1. Define and contrast deterministic and stochastic grammars, providing examples to show the adequacy of each.

2. Identify the classic parsing algorithms for parsing natural language.
3. Defend the need for an established corpus.
4. Give examples of catalog and look up procedures in a corpus-based approach.
5. Articulate the distinction between techniques for information retrieval, language translation, and speech recognition.

IS8. Machine learning and neural networks [elective]

Topics:

- Definition and examples of machine learning
- Supervised learning
- Learning decision trees
- Learning neural networks
- Learning belief networks
- The nearest neighbor algorithm
- Learning theory
- The problem of overfitting
- Unsupervised learning
- Reinforcement learning

Learning objectives:

1. Explain the differences among the three main styles of learning: supervised, reinforcement, and unsupervised.
2. Implement simple algorithms for supervised learning, reinforcement learning, and unsupervised learning.
3. Determine which of the three learning styles is appropriate to a particular problem domain.
4. Compare and contrast each of the following techniques, providing examples of when each strategy is superior: decision trees, neural networks, and belief networks..
5. Implement a simple learning system using decision trees, neural networks and/or belief networks, as appropriate.
6. Characterize the state of the art in learning theory, including its achievements and its shortcomings.
7. Explain the nearest neighbor algorithm and its place within learning theory.
8. Explain the problem of overfitting, along with techniques for detecting and managing the problem.

IS9. AI planning systems [elective]

Topics:

- Definition and examples of planning systems
- Planning as search
- Operator-based planning
- Propositional planning
- Extending planning systems (case-based, learning, and probabilistic systems)
- Static world planning systems
- Planning and execution
- Planning and robotics

Learning objectives:

1. Define the concept of a planning system.
2. Explain how planning systems differ from classical search techniques.
3. Articulate the differences between planning as search, operator-based planning, and propositional planning, providing examples of domains where each is most applicable.
4. Define and provide examples for each of the following techniques: case-based, learning, and probabilistic planning.
5. Compare and contrast static world planning systems with those need dynamic execution.
6. Explain the impact of dynamic planning on robotics.

IS10. Robotics [elective]

Topics:

Overview

- State-of-the-art robot systems
- Planning vs. reactive control
- Uncertainty in control
- Sensing
- World models

Configuration space

Planning

Sensing

Robot programming

Navigation and control

Learning objectives:

1. Outline the potential and limitations of today's state-of-the-art robot systems.
2. Implement configuration space algorithms for a 2D robot and complex polygons.
3. Implement simple motion planning algorithms.
4. Explain the uncertainties associated with sensors and how to deal with those uncertainties.
5. Design a simple control architecture.
6. Describe various strategies for navigation in unknown environments, including the strengths and shortcomings of each.
7. Describe various strategies for navigation with the aid of landmarks, including the strengths and shortcomings of each.

Appendix B

Test Bank

Test Bank

1: (p.2) What does AI stand for?

Chapter 1: Introduction

Chapter 2: Syllabus Extracts

Chapter 3: Readings

Chapter 4: Conditional Probability

2: (p.40) Find $p(A \cap \bar{B})$ given $p(A)=1/3$, $p(B)=7/18$, $p(A \cap B)=1/9$.

3: (p.40) Find $p(\bar{A} \cap B)$ given $p(A)=1/3$, $p(B)=7/18$, $p(A \cap B)=1/9$.

-
- 4: (p.40) Find $p(\overline{A} \cap \overline{B})$ given $p(A)=1/3$, $p(B)=7/18$, $p(A \cap B)=1/9$.
- 5: (p.40) Find $p(A|B)$ given $p(A)=1/3$, $p(B)=7/18$, $p(A \cap B)=1/9$.
- 6: (p.40) Find $p(A|\overline{B})$ given $p(A)=1/3$, $p(B)=7/18$, $p(A \cap B)=1/9$.
- 7: (p.41) Find $p(B|A)$ given $p(A)=1/3$, $p(B)=7/18$, $p(A \cap B)=1/9$.
- 8: (p.41) Find $p(B|\overline{A})$ given $p(A)=1/3$, $p(B)=7/18$, $p(A \cap B)=1/9$.
- 9: (p.41) Find $p(\overline{A} \cap B)$ given $p(A)=5/6$, $p(B)=2/3$, $p(A \cap B)=7/12$.
- 10: (p.41) Find $p(\overline{A} \cap \overline{B})$ given $p(A)=5/6$, $p(B)=2/3$, $p(A \cap B)=7/12$.
- 11: (p.41) Find $p(A|B)$ given $p(A)=5/6$, $p(B)=2/3$, $p(A \cap B)=7/12$.
- 12: (p.41) Find $p(A|\overline{B})$ given $p(A)=5/6$, $p(B)=2/3$, $p(A \cap B)=7/12$.
- 13: (p.41) Find $p(B|A)$ given $p(A)=5/6$, $p(B)=2/3$, $p(A \cap B)=7/12$.
- 14: (p.41) Find $p(B|\overline{A})$ given $p(A)=5/6$, $p(B)=2/3$, $p(A \cap B)=7/12$.
- 15: (p.41) Find $p(A \cap \overline{B})$ given $p(A)=3/7$, $p(B)=3/7$, $p(A \cap B)=2/21$.
- 16: (p.41) Find $p(\overline{A} \cap B)$ given $p(A)=3/7$, $p(B)=3/7$, $p(A \cap B)=2/21$.
- 17: (p.42) Find $p(\overline{A} \cap \overline{B})$ given $p(A)=3/7$, $p(B)=3/7$, $p(A \cap B)=2/21$.
- 18: (p.42) Find $p(A|B)$ given $p(A)=3/7$, $p(B)=3/7$, $p(A \cap B)=2/21$.
- 19: (p.42) Find $p(A|\overline{B})$ given $p(A)=3/7$, $p(B)=3/7$, $p(A \cap B)=2/21$.
- 20: (p.42) Find $p(B|A)$ given $p(A)=3/7$, $p(B)=3/7$, $p(A \cap B)=2/21$.
- 21: (p.42) Find $p(B|\overline{A})$ given $p(A)=3/7$, $p(B)=3/7$, $p(A \cap B)=2/21$.

Chapter 5: Bayesian Probability

- 22: (p.44) What is the product rule?
- 23: (p.44) What is Bayes' rule?

Chapter 6: Propositional Resolution

24: (p.51) Resolve: $(\neg a \ c) (\neg b \ a) (\neg c \ a) (\neg c \ b) (b \ c)$

25: (p.51) Resolve: $(\neg a \ b) (\neg b \ c) (a \ b)$

26: (p.51) Resolve: $(\neg c \ \neg d \ a) (a \ b \ d) (a \ c) (a \ c \ d)$

27: (p.51) Resolve: $(\neg b \ \neg c \ a) (\neg c \ b \ d) (\neg d \ a \ b)$

28: (p.51) Resolve: $(\neg a \ d) (\neg b \ c) (\neg c \ a) (\neg d \ a \ c) (a \ b)$

29: (p.51) Resolve: $(\neg b \ a) (\neg b \ a \ c) (\neg c \ \neg d \ a) (a \ b \ c)$

30: (p.51) Resolve: $(\neg a \ \neg b \ c) (\neg b \ \neg c \ d) (\neg b \ a)$

31: (p.51) Resolve: $(\neg a \ \neg c \ b) (\neg a \ \neg d \ c) (\neg b \ d) (\neg c \ a \ b)$

32: (p.51) Resolve: $(\neg c \ \neg d \ a) (\neg d \ b \ c) (a \ b) (a \ c \ d) (c \ d)$

33: (p.51) Resolve: $(\neg a \ \neg d \ c) (\neg d \ a) (a \ b)$

Chapter 7: Projects in General

Chapter 8: Vacuum

Chapter 9: Hunt the Wumpus

Chapter 10: Number Recognition

Chapter 11: Phonetics

Chapter 12: Exam Topics

- 34: (p.73) Discuss: Artificial Intelligence
- 35: (p.73) Discuss: Turing test
- 36: (p.73) Discuss: automated reasoning
- 37: (p.73) Discuss: machine learning
- 38: (p.73) Discuss: total Turing test
- 39: (p.74) Discuss: agent
- 40: (p.74) Discuss: dualism
- 41: (p.74) Discuss: materialism
- 42: (p.74) Discuss: logical positivism
- 43: (p.74) Discuss: algorithm
- 44: (p.74) Discuss: Godel's incompleteness theorem
- 45: (p.74) Discuss: intractability
- 46: (p.74) Discuss: NP-completeness
- 47: (p.74) Discuss: machine evolution
- 48: (p.74) Discuss: genetic algorithms
- 49: (p.74) Discuss: expert systems
- 50: (p.74) Discuss: frames
- 51: (p.74) Discuss: rational agent
- 52: (p.75) Discuss: autonomous agent
- 53: (p.75) Discuss: simple reflex agent
- 54: (p.75) Discuss: goal-based agent
- 55: (p.75) Discuss: utility-based agent

-
- 56:** (p.75) Discuss: accessible environment
- 57:** (p.75) Discuss: deterministic environment
- 58:** (p.75) Discuss: episodic environment
- 59:** (p.75) Discuss: static vs dynamic environment
- 60:** (p.75) Discuss: discrete vs continuous environment
- 61:** (p.75) Discuss: search
- 62:** (p.75) Discuss: path cost
- 63:** (p.75) Discuss: breadth-first search
- 64:** (p.75) Discuss: uniform-cost search
- 65:** (p.75) Discuss: depth-first search
- 66:** (p.76) Discuss: depth-limited search
- 67:** (p.76) Discuss: iterated deepening search
- 68:** (p.76) Discuss: bidirectional search
- 69:** (p.76) Discuss: heuristics
- 70:** (p.76) Discuss: best-first search
- 71:** (p.76) Discuss: greedy search
- 72:** (p.76) Discuss: A* search
- 73:** (p.76) Discuss: admissible heuristic
- 74:** (p.76) Discuss: knowledge representation
- 75:** (p.76) Discuss: inference (sound, complete)
- 76:** (p.76) Discuss: propositional logic
- 77:** (p.76) Discuss: first-order logic
- 78:** (p.76) Discuss: atomic sentence
- 79:** (p.76) Discuss: predicate

-
- 80:** (p.77) Discuss: quantified sentence
- 81:** (p.77) Discuss: situational calculus
- 82:** (p.77) Discuss: diagnostic rules
- 83:** (p.77) Discuss: causal rules
- 84:** (p.77) Discuss: unification
- 85:** (p.77) Discuss: Modus Ponens
- 86:** (p.77) Discuss: Horn form
- 87:** (p.77) Discuss: resolution
- 88:** (p.77) Discuss: conjunctive normal form
- 89:** (p.77) Discuss: implicative normal form
- 90:** (p.77) Discuss: conditional plans
- 91:** (p.77) Discuss: execution monitoring
- 92:** (p.77) Discuss: action monitoring
- 93:** (p.77) Discuss: replanning agent
- 94:** (p.77) Discuss: prior probabilities
- 95:** (p.77) Discuss: conditional probabilities
- 96:** (p.78) Discuss: joint probability distribution
- 97:** (p.78) Discuss: Bayes' rule
- 98:** (p.78) Discuss: conditional independence
- 99:** (p.78) Discuss: Bayesian updating
- 100:** (p.78) Discuss: belief networks
- 101:** (p.78) Discuss: stochastic simulation
- 102:** (p.78) Discuss: truth-functional system
- 103:** (p.78) Discuss: performance element

-
- 104: (p.78) Discuss: learning element
 - 105: (p.78) Discuss: inductive learning
 - 106: (p.78) Discuss: neural network
 - 107: (p.78) Discuss: perceptron
 - 108: (p.78) Discuss: linearly separable function
 - 109: (p.78) Discuss: feed-forward network
 - 110: (p.78) Discuss: back-propagation
 - 111: (p.78) Discuss: Bayesian learning
 - 112: (p.79) Discuss: multi-layer feed-forward network
 - 113: (p.79) Discuss: speech act
 - 114: (p.79) Discuss: phrase-structure grammar
 - 115: (p.79) Discuss: context-free grammar
 - 116: (p.79) Discuss: encoded message
 - 117: (p.79) Discuss: situated language
 - 118: (p.79) Discuss: augmented grammar
 - 119: (p.79) Discuss: pragmatic interpretation
 - 120: (p.79) Discuss: disambiguation
 - 121: (p.79) Discuss: anytime algorithm
 - 122: (p.79) Discuss: bounded optimality
 - 123: (p.79) Discuss: prisoner's dilemma

Chapter A: CC2001: Intelligent Systems

Chapter B: Test Bank

Index

- 2151, 9
- a priori probabilities, 44
- academic integrity, 26
- ACM, 80
- attendance, 13
- bake off, 54
- basis, 68
- Bayes' rule, 43
- calendar, 11
- CNF, 48, 50
- complementary literals, 49
- conjunction, 39, 48–50
- conjunctive normal form, 50
- course homepage, 10
- CSLU, 65
- daily update, 13
- disjunction, 39, 48, 50
- dress and grooming, 30
- email, 5
- faculty info, 9
- first order logic, 47
- formal communication, 20
- grading, 12
- IEEE-CS, 80
- induction, 68
- informal communication, 20
- learning framework, 16
- learning outcomes, 22
- LMS, 10
- modus ponens, 48
- numbers, 64
- office hours, 9
- over training, 69
- percept, 56
- phonemes, 64
- plagiarism, 27
- Predicate Calculus, 47
- prior probabilities, 44
- probability, 36
- product rule, 43
- Propositional Logic, 47
- resolution, 48
- Robinson, John Alan, 48
- rules of inference, 47
- sexual misconduct, 29
- special needs, 31
- syllogisms, 48
- tardiness, 13
- tautology, 49
- textbook, 10
- tutoring, 18
- vacuum, 55

Venn diagram, [37](#), [40](#)

Wumpus, [60](#)