



Computer Science (CSCI) 698 Practicum in Teaching Computer Science

*(some slides from Laurent Itti
Gaurav S. Sukhatme, Saty R)*

Andrew Goodney
goodney@usc.edu

USC

School of Engineering

University of Southern California



Measuring learning: assessment

- **Summative assessment: of tests, papers, assignments etc.**
 - The goal of summative assessment is to evaluate student learning at the end of an instructional unit by comparing it against some standard or benchmark.
- **Formative assessment: feedback on the nature and progress of student learning**
 - The goal of formative assessment is to monitor student learning to provide ongoing feedback that can be used by instructors to improve their teaching and by students to improve their learning. More specifically, formative assessments: help students identify their strengths and weaknesses and target areas that need work.
- **Summative methods are normally used for assigning grades**
 - Today's material from CET Teaching Nuggets section 4



Assessments should

- be linked to course objectives
- be regular, on-going
- be designed to facilitate learning
- result in feedback to students

CS 561: Artificial Intelligence



Instructors: Profs. Laurent Itti (itti@usc.edu), Wei-Min Shen (shen@isi.edu),
Sheila Tejada (stejada@usc.edu) & Ning Wang (nwang@ict.usc.edu)

TAs: Chi-An Chen - chianc@usc.edu (50%)
Thomas Collins - collinst@usc.edu (50%)
Daniel Link - dlink@usc.edu (25%)
Qian Wang - wang215@usc.edu (50%)
Wuxuan Jiang - wuxuanji@usc.edu (50%)
Kan Qi - kqi@usc.edu (50%)
Bo Wang - wang736@usc.edu (50%)
Ali Kazemian - akazemia@usc.edu (50%)
John Tran - jtran@isi.edu

Lectures: M-W 17:00 – 18:20, – or – Tues. 18:40 – 21:20, SGM-123

Office hours: Mon 13:00 – 14:00, HNB-SGM-123 07A (Prof. Itti)

Discussion: Profs. Shen, Tejada & Wang

This class will use **courses.uscdcn.net** (Desire2Learn, **D2L**)

- Up to date information
- Lecture notes
- Homeworks posting and submission information
- Grades
- Relevant dates, links, etc.

Textbook: [AIMA] Artificial Intelligence: A Modern Approach, by Russell & Norvig. (3rd ed)

Optional (ALFE): Autonomous Learning from the Environment by Shen

CS 561: Artificial Intelligence



Course overview: foundations of symbolic intelligent systems. Agents, search, problem solving, logic, representation, reasoning, symbolic programming, and robotics.

Prerequisites: CS 455x, i.e., programming principles, discrete mathematics for computing, software design and software engineering concepts. **Good knowledge of C++ and STL, or Java, or Python needed for programming assignments.**

Grading: 20% for midterm-1 +
20% for midterm-2 +
30% for final +
30% for 3 mandatory homeworks/assignments

USC

School of Engineering

University of Southern California

CS 561: Artificial Intelligence



Grading:

Grading is absolute and according to the following scale:

≥ 90 A+ (honorary – shows as A on transcript)

≥ 80 A

≥ 75 A-

≥ 70 B+

≥ 60 B

≥ 55 B-

≥ 50 C+

≥ 40 C

≥ 35 C-

< 35 F

USC

School of Engineering

University of Southern California

Policies



- **Class mailing list:** will be setup on the D2L system
- **Homeworks:** See class web page on D2L. Homeworks are programming assignments.
 - Aug 28 – HW1 out Topic: search
 - Sep 20 – HW1 due
 - Sep 25 – HW2 out Topic: game playing or constraint satisfaction
 - Oct 16 – HW2 due
 - Oct 18 – HW3 out Topic: logic reasoning and inference or neural networks
 - Nov 20 – HW3 due
- **Late homeworks:** you lose 20% of the homework's grade per 24-hour period that you are late. Beware, the penalty grows very fast: $\text{grade} = \text{points} * (1 - n * 0.2)$ where n is the number of days late ($n=0$ if submitted on time, $n=1$ if submitted between 1 second and 24h late, etc).
- **Homework grading:** your hws will be graded by an A.I. agent (given to you in advance for testing) through the online system at **vocareum.com**.
- **Grade review / adjustment:** Requests will be considered up to 2 weeks after the grade is released. After that, it will be too late and requests for grading review will be denied.
- **Exams:**
 - Monday, September 25, 8:00pm – 9:50pm – midterm 1 (room TBA)
 - Monday, October 30, 8:00pm – 9:50pm – midterm 2 (room TBA)
 - Friday, December 8, 8:00am – 10:00am – final (room TBA)



Assignments and homework

- Be clear and specific about goals and expectations for each assignment
- Nature and construction of each assignment should reflect the learning goals of the class
- Assignments should be challenging but not intimidating or overwhelming



Useful tips for practice

- Present each assignment verbally and in writing
- Clearly outline expectations
- Test the assignment on yourself and others
- Consider having multiple, shorter assignments
- Consider separating the tasks for younger learners
- Create realistic problems
- Track where students succeed and struggle



More tips

- Coordinate homework with lectures/readings
- Create a reasonable, even load of homework
- Minimize the stakes of any one assignment/due date
- Make the first homework a review
- Balance routine and challenging problems
- Vary the type of homework you assign
- Be prompt in returning homework



Constructing tests

- Create new tests each time you teach a course
- Leave yourself time to write the test
 - What content do you want the students to know?
 - What do you want them to be able to do with the content (recall, discriminate, analyze, etc.)?
- Create a bank of questions during the term
- Pay attention to the layout of the exam



Types of questions

- Objective (1 min/question)
- Short-answer questions (2-5 min/question)
- Essays (30 min for a 2 pager)
- Problems (10 min if it takes you 2 min)
- Performance tests
- Take home exams



After the exam

- Hand out a rubric
- Discuss the exam
- Return graded exams as soon as possible (within a week is advisable)
- Set a period (say a week) when students can discuss and dispute their scores



Assigning grades

- Grades should reflect course goals
- Grading standards should be stated early
- Grade consistently
- Assign grades only on academic performance
- Individual assessments should be weighted according to their importance



Teaching large classes

- Overarching remark and strategy:
- any change to the original plan WILL have (perceived or real) adverse effects onto some students
- hence, they are best accepted if
 - students are allowed to proceed according to original plan (if applicable) and maybe receive a bonus
 - several possible answers are accepted as correct
 - generally speaking, those who have expanded effort going with the original plan should get rewarded for their effort, those who have not yet started should not get penalized by the change.

Example: a student asks for lenient grading but has no case



- “Rubric says correct answer is ABCD. I answered ABC and I just forgot to write down D even though I had it in my mind, can you please give me full credit?”
- should we accept the request?

Example: a student asks for lenient grading but has no case



- “Rubric says correct answer is ABCD. I answered ABC and I just forgot to write down D even though I had it in my mind, can you please give me full credit?”
- Generally speaking: no
 - Rules were well stated in advance
 - You knew what to expect
 - Rubric is correct
 - You made a mistake (always remain courteous and it is ok to be sympathetic as well, e.g.: “I am sorry as indeed this looks like a small omission, but to be fair to everyone we have to stick to the rubric since it is correct”).
 - Generally speaking, you do not want to set a precedent that tells your students they can get marks for incorrect answers.



Focus today:

- What about cases where students do have a case (even if it is weak)?

Example: students request more lenient grading on exam 1



- Exam 1 was tough. Class average is only 59%
- should we accept the request?

Example: students request more lenient grading on exam 1



- Problems:
 - students who comparatively do better on other exams feel penalized
 - students who did well on this exam feel penalized (they are losing their edge)
 - students who did poorly feel penalized (they are not gaining as much)

Example: students request more lenient grading on exam 1

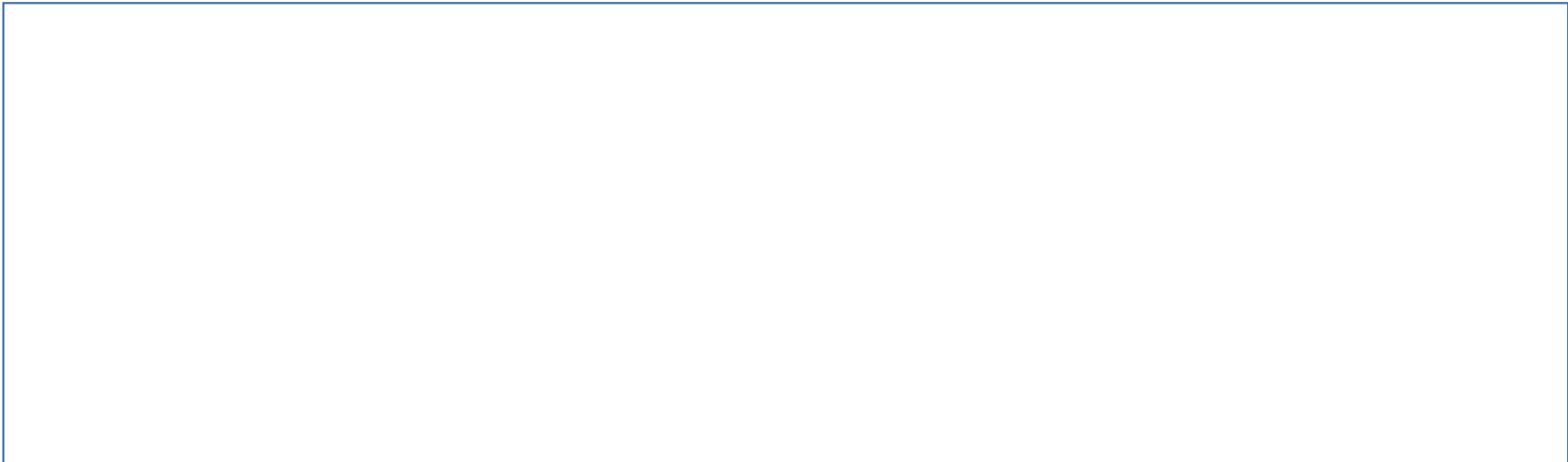


- Possible strategy:
 - Firm: grade policy was clearly defined in advance
 - Lenient: no change for exam 1 but we will make exam 2 easier

2. [10%] FOL Resolution

Consider the following logical sentence in CNF form. Show that this sentence is unsatisfiable.

$$(\neg A \vee \neg B \vee C) \wedge (\neg A \vee B) \wedge (A) \wedge (\neg A \vee \neg B \vee \neg C)$$



? question ☆

174 views

Actions ▾

Final Q2

Question 2 didn't specify that resolution should be used, and I explained the results using a truth table including an explanation of how a contradiction occurs when assigning wrong values. I think it's not fair that I lost 5 points despite my analysis, and 5 points makes a lot of difference for me right now. Please reconsider.

exam

edit

· good question | 7

Updated 9 months ago by [redacted] (anon. to classmates)

i the instructors' answer, *where instructors collectively construct a single answer*

Click to start off the wiki answer

followup discussions *for lingering questions and comments*

Resolved Unresolved



[redacted] 9 months ago

I did the same haha. I used truth table values to show values for A, B, and C that would make the sentence false. The instructions didn't say to use resolution so I didn't think to use resolution.



[redacted] (anon. to classmates) 9 months ago Thanks for your comment, this helps show that the problem statement was not clear enough.

Reply to this followup discussion

Resolved Unresolved



[redacted] (anon. to classmates) 9 months ago

Yes, there is no mention of using resolution to solve it. Deducting 5 marks for no mistake of ours is too bad

Reply to this followup discussion

Resolved Unresolved



[redacted] (anon. to classmates) 9 months ago

I completely agree. This question should be regraded.

Reply to this followup discussion

Resolved Unresolved



[redacted] 9 months ago

the problem statement itself does not mention anything about it demand us to use resolution.

Quite tricky question.

I think this question was given to low the average. It has not any relationship whether u have a good understanding of AI knowledge itself.

Reply to this followup discussion

Resolved Unresolved



[redacted] (anon. to classmates) 9 months ago

The Title is FOL Resolution....



[redacted] (anon. to classmates) 9 months ago If u think the matter in this way.

The topper title is Artificial intelligence. So why could we use AI method.

Reply to this followup discussion



Possible strategy:

- Accept truth table solution as well
- But only with partial credit, or give a bonus to resolution solutions, since title of question did mention resolution, and CNF was only used in class with resolution



Exam design: mistake

6. [10%] AI Applications

Based on material from discussion, circle the **best** choice for each question:

- (e) [2%] If h_1 and h_2 are admissible heuristics, which of the following are also guaranteed to be admissible?
- a. $h_1 * h_2$
 - b. $\max(h_1, h_2)$
 - c. $(\alpha)h_1 + (1-\alpha)h_2$, for $\alpha \in [0,1]$
 - d. All of the above
 - e. None of the above



Exam design: mistake

- (e) [2%] If h_1 and h_2 are admissible heuristics, which of the following are also guaranteed to be admissible?
- a. $h_1 * h_2$
 - b. $\max(h_1, h_2)$
 - c. $(\alpha)h_1 + (1-\alpha)h_2$, for $\alpha \in [0,1]$
 - d. All of the above
 - e. None of the above

ooops, b and c are both correct



Exam design: mistake

- (e) [2%] If h_1 and h_2 are admissible heuristics, which of the following are also guaranteed to be admissible?
- a. $h_1 * h_2$
 - b. $\max(h_1, h_2)$
 - c. $(\alpha)h_1 + (1-\alpha)h_2$, for $\alpha \in [0,1]$
 - d. All of the above
 - e. None of the above

Better solution: accept either answer or both

give extra credit if student circled both or wrote something about both being correct



Homework design: mistake

Strategy:

- extend deadline
- but allow on time submissions and give a bonus to those

USC

School of Engineering

University of Southern California



Deadline extension requests

Decide before the course starts how much time you want to spend adjudicating deadline extension requests...

- Suggestion: no extensions
 - Except for bona fide medical requests beyond “not feeling well” aka a cold or similar
- If you are lenient, you will get EVERY excuse in the book
- Construct deadlines/late penalties to give students flexibility to be a little late with only a small penalty



Deadlines/Tech “issues”

I finished the CS 103 Blackjack Project approximately September 15 and toggled it to Completed but recently turned it back to view my code (I didn't edit anything). Unfortunately, I did not toggle it back to Completed until 30 mins after the deadline. Is there any way to prove that I had it completed before the deadline and not get any late points? thanks

[Comment](#) [Edit](#) [Delete](#) [Endorse](#) ***

Hi! I marked project 1 as complete this weekend; however, when logging in to Codio today, it was no longer marked as complete. Can the late penalty please be removed? I had already passed all the tests for the project days prior to the deadline, which can be seen in the last test cases. Please let me know and thank you. I apologize for the inconvenience.

[Comment](#) [Edit](#) [Delete](#) [Endorse](#) ***

USC

School of Engineering

University of Southern California



Exam Regrade Requests

Q1.10

2 Points

The following code is very similar to something I saw in office hours:

```
Product* makeBook(/* useful arguments */)
{
    Book myBook(/*args*/);
    return &myBook;
}
```

The code compiles, but has an issue. Briefly describe the problem:

the problem is that the function returns a local variable (myBook) that goes out of scope.

Rewrite the function to be correct (it can be a one-liner):

```
Book myBook = new Book(/*args*/);
```

Previous Requests

Student Request

I still do believe that my answer is correct, and goes into the alternate correct so deserves an extra .5 for dynamically allocating myBook.

Staff Response [\(edit\)](#)

you did not make a book pointer or a return statement

USC

School of Engineering

University of Southern California



Exam Regrade Requests

Q2.10
1 Point

Using function _____ allow the programmer to declare functions in any order.

declaration

Previous Requests

Student Request

For Q2.10, I knew and meant it was prototype but I guess a declaration also makes sense because it is kind of a declaration. Please clarify if I am wrong.

Thank you.

Staff Response [\(edit\)](#)

prototype is only right.



Teaching and Tools

- Teaching has always involved tools
- Lectures
 - Chalkboard -> overheads -> PPT...
- Labs
 - Lots of tools in chem/physics labs
- Exams
 - Printed exams, Scantrons



Technology for teaching

- Teaching CS uses lots of tech tools
 - LMS: blackboard / Brightspace
 - Exams: Crowdmark, Gradescope
 - Discussion: Inside LMS, Piazza, EdStem
 - Coding: Vocareum, Codio
 - Plagiarism: MOSS, JPLAG
- Laptops/PCs
- Networks
- Hardware
 - CPU trainers
 - IoT



Emerging tools

- Generative AI
 - ChatGPT



CS Teaching Tech

- What tech to use?
 - University provided?
 - How will students access?
 - Free or some cost?
 - How will it make your life easier?
- How much time will it take?
 - Content creation (slides, exams, assignments)
 - Delegation/Collaboration?
- Lock-in?
 - Content import/export
 - Beware bespoke solutions!



Developing Tools

- Developing tools is CS education research
 - Track at SIGCSE for tools
- Share good tools with colleagues