**Logic for Ontologies**

PHI 637 LEC

Fall 2024

University at Buffalo

**Instructor**: John Beverley, Department of Philosophy

**Office**: 117 Park Hall

**Office Hours**: Tuesdays, 9-11am

**Email**: [johnbeve@buffalo.edu](mailto:johnbeve@buffalo.edu)

**Course Time:** Thursday, 1-3:40pm

**Location**: Park Hall 141

**Course Description**

Humans are perplexing. Many engage effortlessly in discourse without violating conversational norms. Some accurately diagnose treatment options for medical conditions, based on minimal information. Some identify lemmas needed to prove theorems too complex for automated approaches. Humans are *perplexingly* good at solving complicated tasks. Artificial Intelligence communities have, for decades, worked to design computing systems able to solve complicated tasks as well as humans can. Siri, autonomous vehicles, Computer-Aided Diagnosing systems, and automated theorem provers, are examples of the fruits of such labor. For such feats of computing ingenuity to work properly, however, relevant knowledge must be represented in formalisms interpretable by computing systems.

One goal of this course is to provide students with a deep understanding of formalisms underwriting contemporary knowledge representation. We will examine several ‘Description Logics’ which reflect decidable fragments of First-Order Logic and provide formal foundations for widely used semantic web languages. Semantic web languages – such as the Resource Description Framework – in turn provide concrete vocabularies used to represent information across the web. Another goal of this course is to provide students with a deep understanding of these semantic web languages, emphasizing their importance to the development of ontologies – structured vocabularies comprised of human and computer interpretable terminological content representing entities in some domain. Students will gain competency in the application of semantic web languages to represent the philosophical commitments of one of the most important ontologies in the world: Basic Formal Ontology (BFO).

Ontology modeling of this sort is just a first step towards capturing the perplexity of human intelligence. Students will take a further step towards that goal in this course by empirically evaluating the benefit of using ontologies like BFO in the real world. To that end, students will learn to use the Protégé ontology editor to represent BFO hierarchies, automated reasoners native to Protégé to check for logical consistency, the SPARQL semantic web querying language to extract important information from BFO-conformant datasets, and the SHACL semantic web language to validate dynamic updating of BFO-conformant ontologies. Students will also learn to work with subject-matter experts to set up studies aimed at evaluating development workflows, use cases, and ontology engineering metrics. Throughout, students will learn to use GitHub – a common version control environment in the ontology developer toolkit, and in doing so gain insight into how knowledge represented using semantic web standards is revised and maintained across a wide range of stakeholders, users, and contributors.

Will, at the end of this course, students be able to capture the perplexing human ability to solve complex tasks? Probably not. Students will, however, be able to recognize how far contemporary Artificial Intelligence research has progressed towards that goal in an empirically justifiable manner, viewed through the intersecting lenses of logic and ontology.

**Prerequisites**

Students should have either taken UB’s Symbolic Logic 215 or equivalent to earn credit in this course. Certain courses in computer science, linguistics, mathematics, and engineering may satisfy this requirement. If you feel that you might have adequate exposure to symbolic logic to satisfy this requirement, please contact me and we will discuss.

**Required Reading**

All readings will be sent through your “buffalo.edu” email, unless requested otherwise.

**Student Learning Outcomes**

|  |  |  |
| --- | --- | --- |
| **Topics** | **Learning Outcomes** | **Assessments** |
| Description Logic | * Students will gain an understanding of the ontological and practical motivations for and actual implementations of varieties of Description Logics. * Students will gain an understanding of how Description Logics provide the foundation for modern semantic web standards. * Students will learn how to use automated theorem provers to derive consequences from Description Logic implementations. | Pull Request Projects 1, 2, and 3 |
| Protégé | * Students will gain competence in the use of the Protégé ontology editor, as well as various extensions supporting Description Logic querying, SPARQL querying, and graphical visualization. * Students will learn how to build a BFO-conformant ontology using the Protégé editor. | All |
| Basic Formal Ontology & Common Core Ontologies | * Students will gain an understanding of philosophical and practical motivations for, applications of, and developments in Basic Formal Ontology and Common Core Ontologies. * Students understand how Basic Formal Ontology and Common Core Ontologies compare to other ontologies, and be able to speak competently about its strengths and weaknesses. | All |
| W3C Standards | * Students will develop competence in a range of World Wide Web Consortium standards, such as the Resource Description Framework (RDF), RDF Schema (RDFs), and the Web Ontology Language (OWL). * Students will gain an understanding of the SPARQL querying language, used to extract information from ontologies constructed using RDF, RDFs, and OWL. * Students will learn how to validate ontologies as data is added, using the Shapes Constraint Language (SHACL). * Students will gain an understanding of how developer packages are created, through the use of the SHACL Dash library. | Pull Request Projects 3, 4, and 5 |
| Version Control | * Students will gain experience using Github as a version control system for the course. * Students will gain competency in navigating command line and graphical user interface access to Github, will learn to fork, open pull requests, resolve conflicts, merge, etc. | All |
| Study Design | * Students will gain experience setting up empirical studies aimed at evaluating ontology artifacts, ontology engineering methods, and common tools * Students will learn to leverage support from subject-matter experts across UB who specialize in study design, in the interest of empirical research | All |

**Course Requirements**

*Students Seeking Letter Grade*

1. Students seeking a letter grade will be expected to keep up with the readings each week and be actively engaged with peers for in-class group exercises and discussions.
2. Students seeking a letter grade will be expected to complete five projects assigned in class, submit answers to Github for review, help peers refine their own answers, and incorporate peer feedback when refining their own submissions.
3. Students seeking a letter grade will be expected to complete one final project.

Attendance/Participation, Pull Request projects, and the Final Project will be graded on a 100-point scale weighted and scored as follows:

|  |  |
| --- | --- |
| Weighting | Assessment / Assignment |
| 30% | Attendance/Participation |
| 30% | Pull Request Exercises |
| 40% | Final Project |

|  |  |  |
| --- | --- | --- |
| Grade | Quality Points | Percentage |
| A | 4.0 | 93.0% -100.00% |
| A- | 3.67 | 90.0% - 92.9% |
| B+ | 3.33 | 87.0% - 89.9% |
| B | 3.00 | 83.0% - 86.9% |
| B- | 2.67 | 80.0% - 82.9% |
| C+ | 2.33 | 77.0% - 79.9% |
| C | 2.00 | 73.0% - 76.9% |
| C- | 1.67 | 70.0% - 72.9% |
| D+ | 1.33 | 67.0% - 69.9% |
| D | 1.00 | 60.0% - 66.9% |
| F | 0 | 59.9 or below |

*Students Seeking S/U*

1. Students seeking S/U credit will be expected to keep up with the readings each week, be actively engaged with peers for in-class group exercises and participate in class discussions.
2. Students seeking S/U credit will be expected to complete at least four projects assigned in class, submit answers to Github for review, help peers refine their own answers, and incorporate peer feedback when refining their own submissions.
3. To receive S credit for the seminar, students must receive an average of 80 from Attendance/Participation and at least four project submissions.

*Pull Request Projects*

Students will be expected to complete Pull Request project assignments. Completion will require familiarity with contemporary version control systems, in particular, a Github account that is linked to the course Github page where submissions will be submitted.

The nature of Pull Request projects will vary as we proceed through the course. Initial assignments students will take the form of questions inspired by or directly from one or more assigned readings. Later assignments may take the form of constructing an ontology, with logical connections specified, of some area of the world the student finds interesting. Even later assignments will may the form of inputting data about the relevant domain followed by validating that one’s ontology remains consistent through such additions.

Following submission to the course page on Github, students will be assigned the task of reviewing the assignments of other student submissions. The expectation is that reviews *be charitable* and *helpful*. Be courteous. Also, offer substantial criticism or insight. Leaving a comment consisting solely of “I don’t think this is right” or “Great job” is a great way to irritate me. Don’t do that.

**Incompletes**

Only students who have sustained serious illnesses or who have been subject to other sustained,

serious, unforeseeable circumstances that interfere with school work will be allowed to take

incompletes. Students who think they qualify must contact me before the due date for the

seminar paper. We must reach an agreement on the date by which requirements will be

completed before I will grant an incomplete. I enforce all other university requirements for

incompletes.

**E-Mail Announcements**

I will occasionally make announcements to members of this class via e-mail. You will be held

responsible for all such announcements. I will send these messages to students’ e-mail accounts

via UBLearns. In almost all cases, UBLearns sends e-mail messages to students’ buffalo.edu e-

mail accounts. Therefore, it is in your own self-interest to make sure that you receive e-mail from your UB e-mail account.

**Course Conflicts with University-Recognized Commitments**

Those of you who anticipate having a conflict with one or more exams due to religious

observances or membership in college organizations (e.g., sports teams) must let me know as

soon as possible, and preferably no later than the second week of class.

**Academic Integrity**

Academic integrity is a fundamental university value. Through the honest completion of academic work, students sustain the integrity of the university while facilitating the university's imperative for the transmission of knowledge and culture based upon the generation of new and innovative ideas.

Here, you can find the university graduate ([https://www.buffalo.edu/grad/succeed/current-students/policy-library.html#academic-integrity)](https://www.buffalo.edu/grad/succeed/current-students/policy-library.html#academic-integrity) academic integrity policy and any additional instructor requirements and comments regarding academic dishonesty.

**Accessibility Resources**

If you have any disability which requires reasonable accommodations to enable you to participate in this course, please contact the Accessibility Resources, 60 Capen Hall, 645-2608, and the instructor of this course. The office will provide you with information and review appropriate arrangements for reasonable accommodations. <http://www.buffalo.edu/studentlife/who-we-are/departments/accessibility.html>

**Topics and Tentative Schedule**

The following is a list of assigned readings and assignments for each week of the course. You are strongly encouraged to keep up with the readings. I will provide each of the articles listed below online in pdf form or via a link in the table below. The readings are subject to change, though I will make every effort to provide a one-week advanced notice prior to changing a reading assignment.

The readings can be quite dense. To increase the odds of comprehension, I will work through slides each class meeting summarizing main points from the readings that week.

|  |  |  |  |
| --- | --- | --- | --- |
| Week | Topic | Required Reading(s) | Due before Class |
| 1 | *A Brief History of Logic*  *Description Logic*  *Exercise: Git, GitHub, VS Code* | * [Description Logic Primer](https://www.cs.ox.ac.uk/people/ian.horrocks/Publications/download/2014/KrSH14.pdf) | * Install Protégé * Install VS Code * Link GitHub to Class Page |
| 2 | *RDF, RDFs, OWL*  *Direct Semantics*  *Exercise: OWL Cheat Sheet* | * [RDF Primer](https://www.w3.org/TR/rdf11-concepts/) * [RDFS Primer](https://www.w3.org/TR/rdf11-schema/) * [Ch. 4, Keet](https://people.cs.uct.ac.za/~mkeet/files/OEbook.pdf) | * None |
| 3 | *OWL Continued*  *Tableau Reasoning*  *Exercise: OWL Cheat Sheet* | * [OWL 2 Primer](https://www.w3.org/TR/owl2-overview/) * Ch. 12, 13 Allemang | * Open Pull Request for Project #1 |
| 4 | *Development Workflow*  *SPARQL*  *Exercise: Quality Control* | * [Automating Ontology Workflows](https://oboacademy.github.io/obook/lesson/automating-ontology-workflows/) * Ch. 6 Allemang | * Close Pull Request for Project #1 |
| 5 | *Advanced SPARQL*  *Exercise: Quality Control* | * [Learn SPARQL](https://sparql.dev/) |  |
| 6 | *Insecure SPARQL*  *Exercise: Quality Control* | * [Learn SPARQL](https://sparql.dev/) | * Open Pull Request for Project #2 |
| 7 | *ONTOBRAS* | * None |  |
| 8 | *Basic Formal Ontology (*[*repo*](https://github.com/BFO-ontology/BFO-2020)*)*  *Common Core Ontologies (*[*repo*](https://github.com/CommonCoreOntology/CommonCoreOntologies)*)*  *Exercise: Study Design* | * Ch. 5, 6 Smith * [The Common Core Ontologies](https://arxiv.org/pdf/2404.17758) | * Close Pull Request for Project #2 |
| 9 | *STIDS* | * None | * Open Pull Requests for Project #3 |
| 10 | *DOLCE (*[*repo*](https://github.com/appliedontolab/DOLCE)*)*  *BORO*  *Exercise: Study Design* | * [DOLCE](https://arxiv.org/pdf/2308.01597) * [BORO](https://www.borosolutions.net/boro-introduction-industrial-application-ontology) |  |
| 11 | *Schema Languages*  *SHACL*  *Exercise: TQC* | * [SHACL Primer](https://www.w3.org/TR/shacl/) * Ch. 7 Allemang | * Close Pull Requests for Project #3 |
| 12 | *Dash Library*  *Expanding Dash*  *Exercise: TQC* | * [DASH Shapes Vocabulary](https://datashapes.org/dash.html) | * Open Pull Requests for Project #4 |
| 13 | *Ontology Metrics*  *Exercise: Study Design* | * [Ontology Evaluation Methods](https://hal.science/hal-01274199/document) | * Close Pull Request for Project #4 |
| 14 | *Ontology Metrics*  *Exercise: Study Design* | * [Evaluating Foundational Ontologies](https://link.springer.com/content/pdf/10.1007/978-3-642-21034-1_22.pdf) | * Open Pull Requests for Project #5 |
| 15 | *Ontology Metrics*  *Exercise: Study Design* | * [Ontology Pitfall Scanner](https://oops.linkeddata.es/index.jsp) * [Ontology Evaluation Survey](https://cobweb.cs.uga.edu/~kochut/teaching/8350/Papers/Ontologies/OntologyEvaluationSurvey.pdf) |  |
| 16 | *Finals* | * None | * Close Pull Requests for Project #5 |