

Computational Law

Introduction

Michael Genesereth
Computer Science Department
Stanford University

Computational Law

The Cop in the Backseat

Michael Genesereth
Computer Science Department
Stanford University

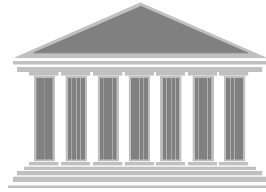
Computational Law

Law Without Lawyers

Michael Genesereth
Computer Science Department
Stanford University



Variety of Regulations



Laws



Business Rules



Personal Rules

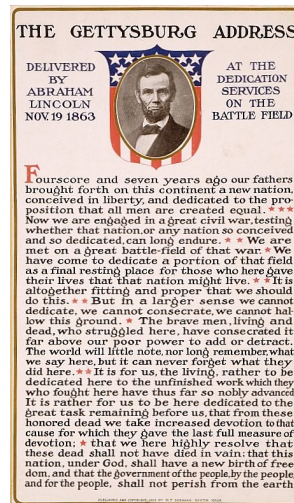


Contracts

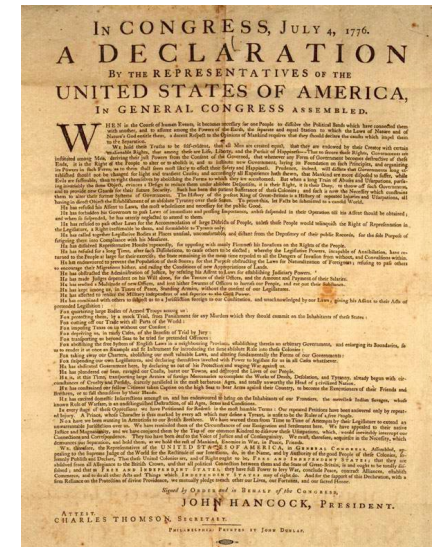
Size of Regulations



66 Words



286 Words



1322 Words



26,911 Words

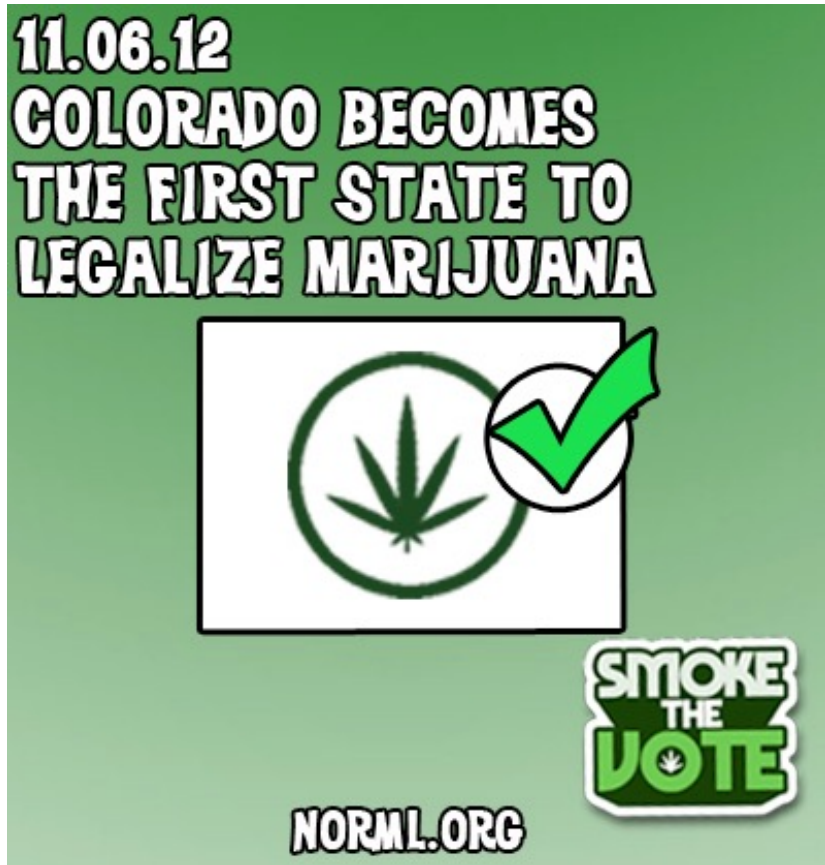
Complexity of Regulations

Excerpts from my homeowner's insurance policy:

Page 32: "*Coverage includes... water damage...*"

Page 112: "*The coverage on page 32 does not apply to damage caused by flood water.*"

Inconsistency



State



Federal

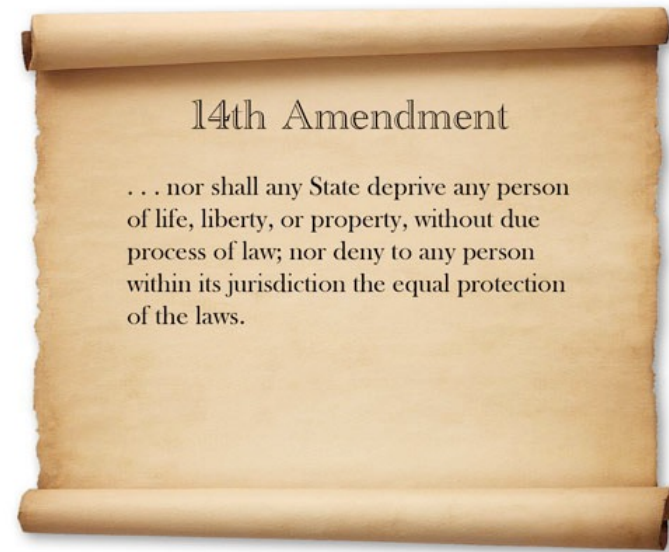
Problems

Problems:

- Variety of Regulations - multiple jurisdictions
- Size of Regulations
- Complexity of Regulations
- Incompleteness and Inconsistency

Results:

- Widespread inefficiency
- Lack of compliance
- Frequent disenchantment with the legal system



It is one of the greatest anomalies of modern times that the law, which exists as a public guide to conduct, has become such a recondite mystery that it is incomprehensible to the public and scarcely intelligible to its own votaries.

- Lee Loevinger 1949

Computational Law

Information Problem



Answer: Legal Technology

Textual Representations of the Law

Good news - most online legal information (e.g. cases, statutes, analysis) online in the form of free-form text, in many cases, with key phrases and pagination.



Bad news - The *bad news* is that this does not provide adequate search and does not support automation.

Computational Law

Computational Law is that branch of legal informatics concerned with **computable** rules and regulations.

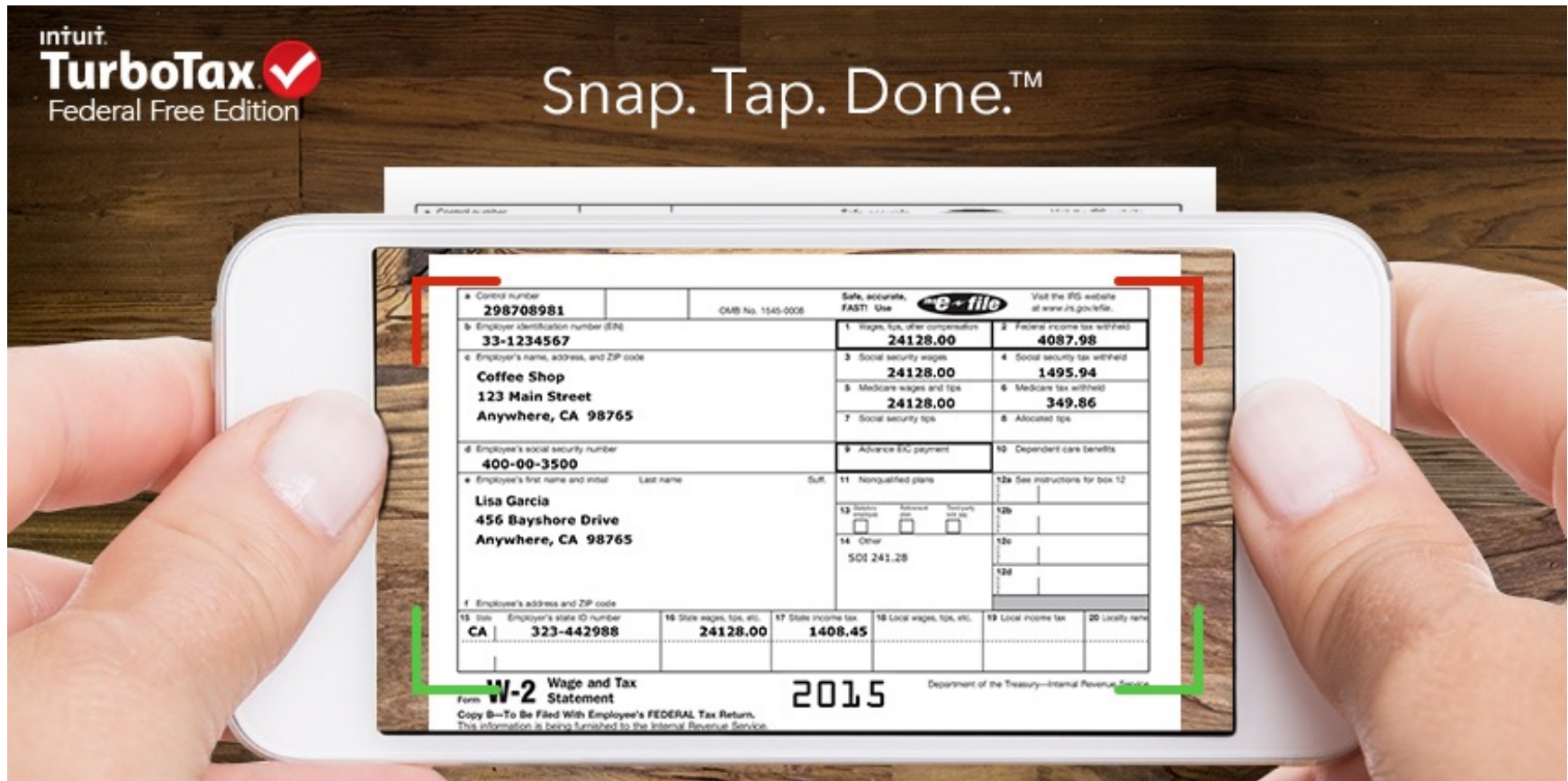


Legal Formalism



LAW in the MACHINE

Turbotax (Intuit)



Portico (Symbium)

Not Secure — complaw.stanford.edu

Portico

Use sliders to adjust view. Click and drag to move building. Click Larger, Smaller, Taller, Shorter to adjust size.

Larger Smaller Turn Stop Taller Shorter

Item	Data
Zone	R-1

Standard	Actual	Allowed	Status
Footprint	160000	168000	✓

Item	Min	Max
Home x	200	600

<http://complaw.stanford.edu/chapters/portico.html>

Turbotaxing the World

Building Codes

Electronic Commerce (transportation, shipping,...)

Privacy (HIPAA, COPPA, etc.)

Labor Law

Financial Management (stocks, retirement funds)

Civil Procedures

California Self Help Guide



The screenshot shows a web browser window with the URL `selfhelp.courts.ca.gov`. The page features a dark blue header with the title "Self Help Guide to the California Courts" and a sub-header "Resources and information to help you navigate your court case, including step-by-step guides for following procedures and help with understanding your options." Below this is a question "What would you like to do?" and six interactive cards, each with a title, description, and a "Go" button.

Self Help Guide to the California Courts

Resources and information to help you navigate your court case, including step-by-step guides for following procedures and help with understanding your options.

What would you like to do?

- Get help with papers I was served**
Look up by form number and understand your options
Choose form [v] **Go**
- Start a court case**
Find your case type to get started
Choose case type [v] **Go**
- Take action on my court case**
Move your case forward, make a change to a decision, dismiss a case
Choose case type [v]
Choose action [v] **Go**
- Get information about a legal matter**
Find out what kind of help you can get from the court for a variety of situations
Choose case type [v] **Go**
- Look up a court case or citation**
Find a traffic ticket or court case using the county court's website
Choose County [v] **Go**
- Get help from the court**
Find in-person self-centers, court interpreters, disability access and more.
Choose resource [v] **Go**

Computable Contracts

Insurance Contracts

Stock Trading Contracts

Real Estate Transactions

Information Access Contracts

Alimony and Child Support Contracts

Academic Program Sheet

DEPARTMENT OF COMPUTER SCIENCE
MSCS Program Sheet (2010-11)

Artificial Intelligence Primary Specialization

Name: **Charles Parnell Naut** Advisor: Proposed date for degree conferral: Date: 10/8/2010
Student ID #: Email: HCP? Coterm?

GENERAL INSTRUCTIONS

Before the end of your first quarter, you should complete the following steps. Detailed instructions are included in the **Guide to the MSCS Program Sheet** in your orientation packet (an online version is available at cs.stanford.edu/degrees/mscs/programsheets/):

- Complete this program sheet by filling in the number, name and units of each course you intend to use for your degree.
- Create a course schedule showing the year and quarter in which you intend to take each course in your program sheet.
- Meet with your advisor and secure the necessary signatures on the program sheet.

FOUNDATIONS REQUIREMENT

You must satisfy the requirements listed in each of the following areas; all courses taken elsewhere must be approved by your adviser on a foundation course waiver form. Required documents for waiving a course include course descriptions, syllabi, and textbook lists. These document can be organized here: cs.stanford.edu/degrees/mscs/waivers/. Do not enter anything in the "Units" column for courses taken elsewhere.

Note: If you are amending an old program sheet, enter "**on file**" in the approval column for courses that have already been approved.

Required:	Equivalent elsewhere (course number/title/institution)	Approval	Grade	Units
Logic, Automata and Complexity (<input checked="" type="checkbox"/> CS 103)	<input type="text"/>	<input type="text"/>	<input type="text"/>	4
Probability (<input type="checkbox"/> CS 109, <input type="checkbox"/> STATS 116, <input type="checkbox"/> CME 106, or <input type="checkbox"/> MS&E 220)	<input type="text"/>	<input type="text"/>	<input type="text"/>	
Algorithmic Analysis (<input checked="" type="checkbox"/> CS 161)	<input type="text"/>	<input type="text"/>	<input type="text"/>	5
Computer Organization and Systems (<input checked="" type="checkbox"/> CS 107)	<input type="text"/>	<input type="text"/>	<input type="text"/>	5
Principles of Computer Systems (<input checked="" type="checkbox"/> CS 110)	<input type="text"/>	<input type="text"/>	<input type="text"/>	5

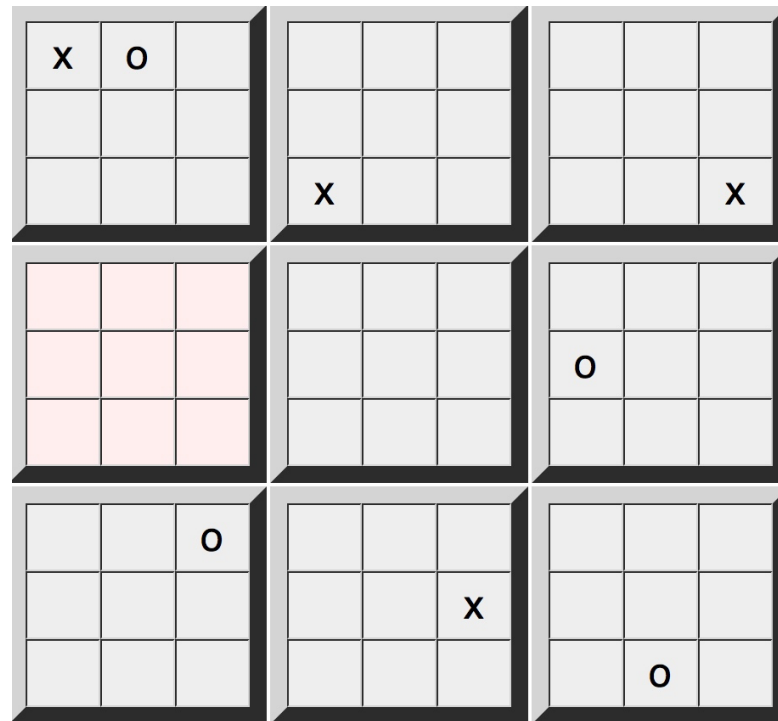
TOTAL UNITS USED TO SATISFY FOUNDATIONS REQUIREMENT:

Note: This total may not exceed 10 units.

7 Requirements Left Total Units: 10 Status: Draft

<http://complaw.stanford.edu/chapters/program.html>

Nineboard Tic Tac Toe



<http://complaw.stanford.edu/chapters/nineboard.html>

Technology

Databases and Knowledge Graphs

Column Name	Condensed Type	Nullable
CITY_ID	int	NOT NULL
CITY_NAME	char(30)	NULL
COUNTRY	char(30)	NULL

Column Name	Condensed Type	Nullable
FLIGHT_ID	int	NOT NULL
FLIGHT_CODE_CHARGE	char(10)	NULL
FROM_CITY_ID	int	NULL
TO_CITY_ID	int	NULL
NUM_OF_SEATS	int	NULL
DEP_TIME	datetime	NULL
ARR_TIME	datetime	NULL
FREQUENCY	char(20)	NULL
TICKET_PRICE	int	NULL

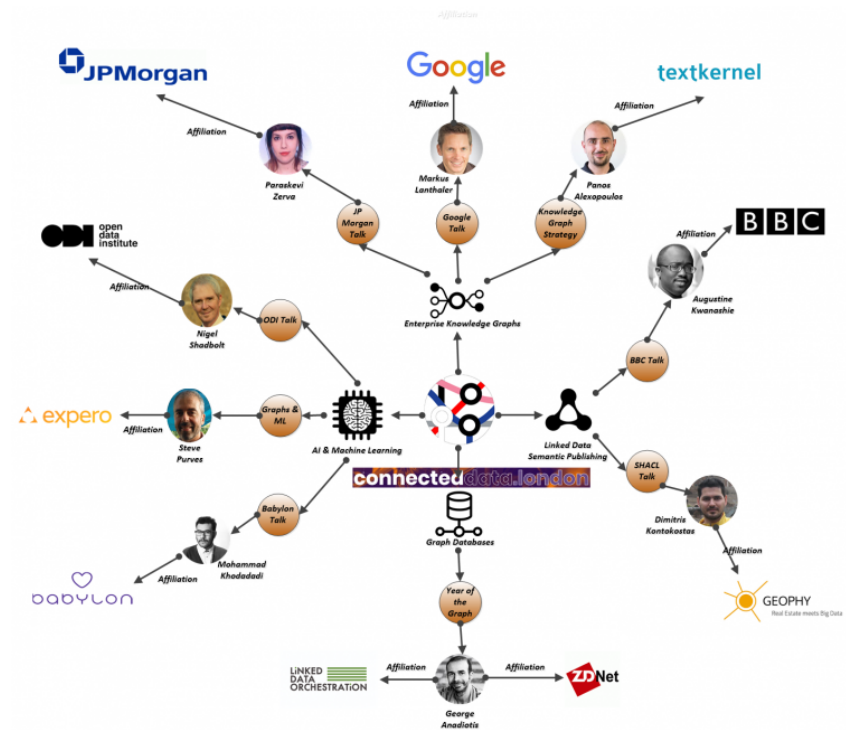
Column Name	Condensed Type	Nullable
CREDIT_CARD_NUM	char(10)	NOT NULL
CREDIT_CARD_TYPE	int	NOT NULL
CREDIT_CARD_MONTH	int	NOT NULL
CREDIT_CARD_YEAR	int	NOT NULL
CREDIT_CARD_BAL	int	NULL

Column Name	Condensed Type	Nullable
RESERVATION_ID	int	NOT NULL
NUM_OF_SEATS	int	NULL
PURCHASED_FLAG	bit	NULL
CREDIT_CARD_NUM	char(30)	NULL
CREDIT_CARD_TYPE	int	NULL
CREDIT_CARD_MONTH	char(10)	NULL
CREDIT_CARD_YEAR	char(10)	NULL
RESERVATION_DATE	datetime	NULL
FIRST_NAME	char(30)	NULL
EMAIL	char(30)	NULL
RESERVATION_NOTIFIED	bit	NULL
PURCHASE_NOTIFIED	bit	NULL
TRANSACTION_ID	bigint	NULL
USER_ID	int	NULL

Column Name	Condensed Type	Nullable
PASSENGER_ID	int	NOT NULL
RESERVATION_ID	int	NOT NULL
FIRST_NAME	char(20)	NULL
LAST_NAME	char(20)	NULL

Column Name	Condensed Type	Nullable
USER_ID	int	NOT NULL
USER_NAME	char(25)	NOT NULL
TRANSACTION_ID	bigint	NULL
TRANSACTION_TYPE	char(10)	NULL

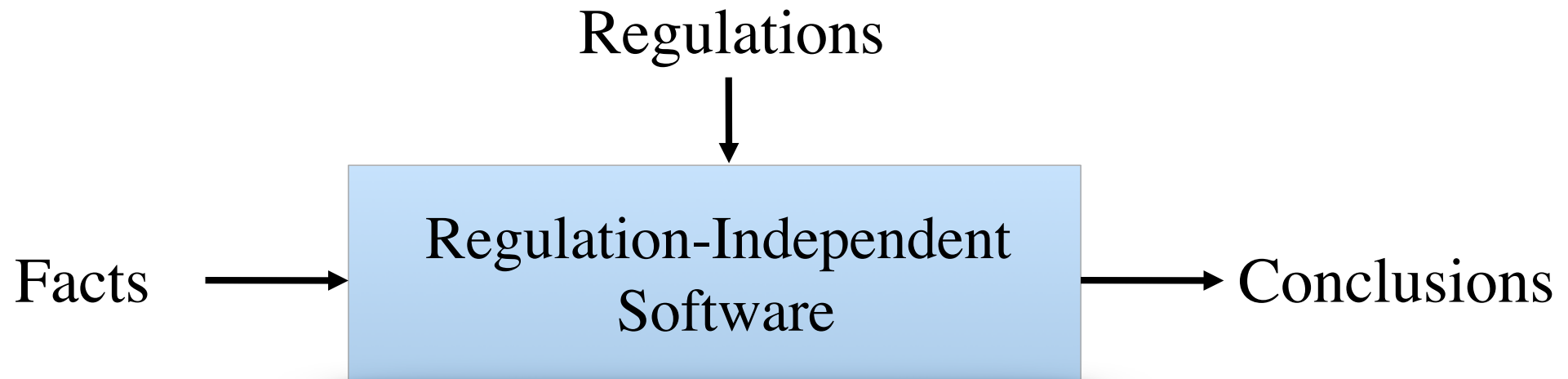
Column Name	Condensed Type	Nullable
INVOICE_ID	int	NOT NULL
RESERVATION_ID	int	NULL
AMOUNT	int	NULL
DATE_PAID	datetime	NULL
PAYMENT_TYPE	char(20)	NULL
CREDIT_CARD_NUM	char(30)	NULL
CREDIT_CARD_TYPE	char(20)	NULL
CREDIT_CARD_MONTH	char(10)	NULL
CREDIT_CARD_YEAR	char(10)	NULL
TRANSACTION_ID	bigint	NULL
USER_ID	int	NULL
ccAuthorized	bit	NULL



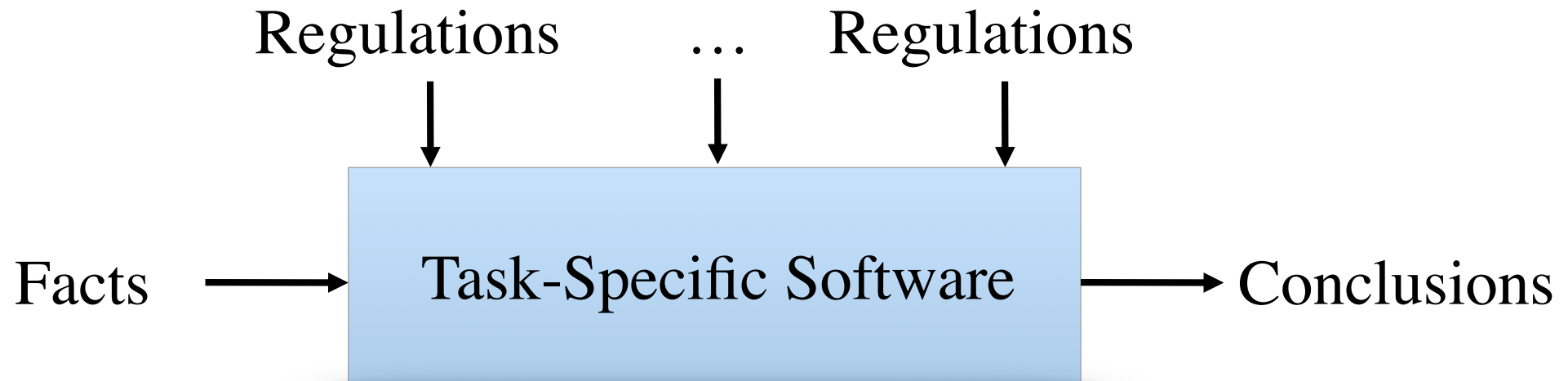
Regulations as Programs



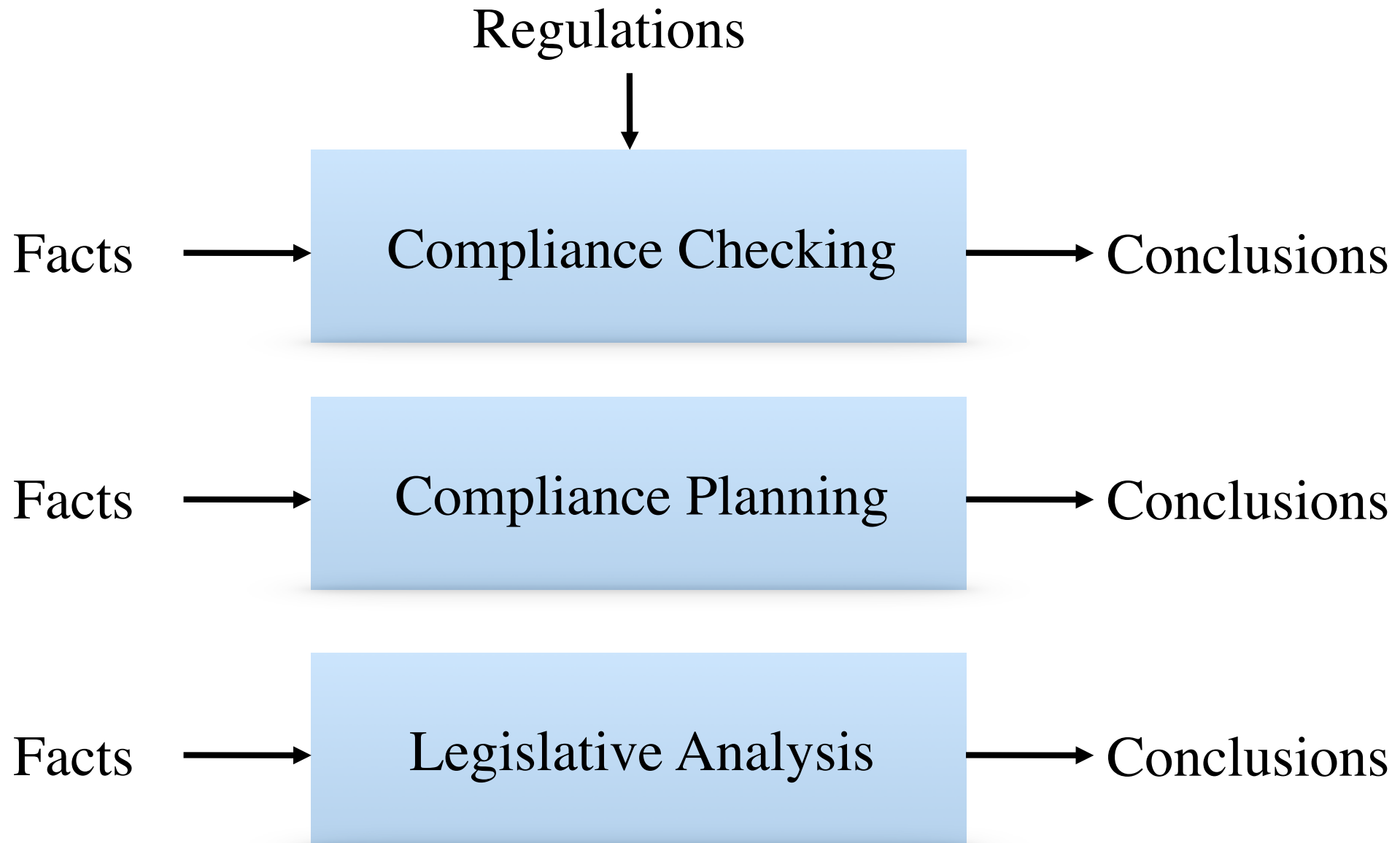
Regulations as Data



One System, Multiple Regulations



One Representation, Multiple Uses



Computational Logic / Logic Programming

Definitions of Concepts

Office mates are people who share an office.

`officemate(X,Y) :- office(X,Z) ^ office(Y,Z)`

Definitions of Legality

Managers and subordinates may not be office mates.

`illegal :- manages(X,Y) ^ officemate(X,Y)`

The new Legalese!

Deductive Reasoning Systems

*Office mates are people who share an office.
Managers and subordinates may not be office mates.*

Automated
Reasoner

*John manages Ken.
John is in 22.
Ken is in 22.*

Violation detected.

Planning for Compliance

*Office mates are people who share an office.
Managers and subordinates may not be office mates.*

Automated
Reasoner

*John manages Ken.
John is in 22.*

Ken not in 22.

Analyzing Regulations

*All projects have managers and subordinates.
No manager may share an office with a subordinate.
All skunkworks personnel must be housed in a common room.*

Automated
Reasoner

Inconsistency detected.

Open Texture



Analogical Reasoning

“Law is not a matter of simply applying rules to facts via Modus Ponens.”

- Edwina Rissland

Many legal decisions are made through case-based / analogical reasoning, bypassing explicit reasoning about laws and statutes.

While there is research on analogical reasoning, there is no widely accepted standard for what constitutes a “correct” answer.

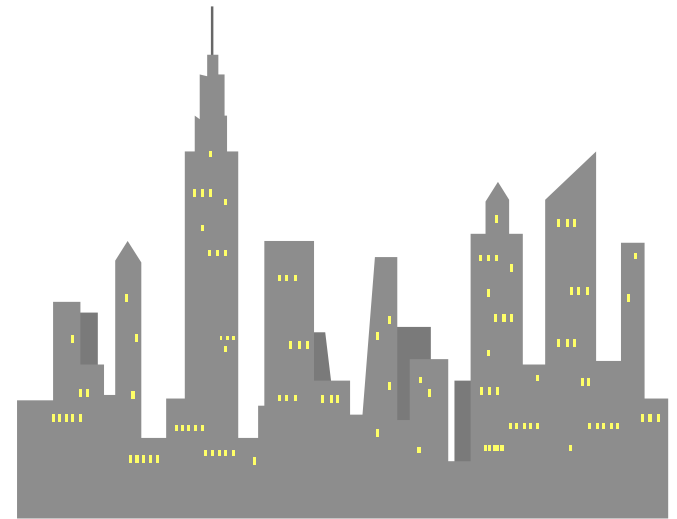
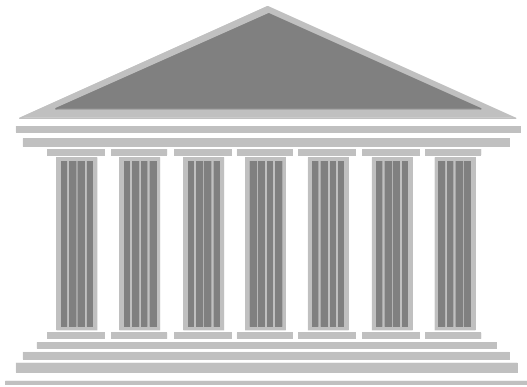
Enterprise Management



ORACLE®



Government as Enterprise

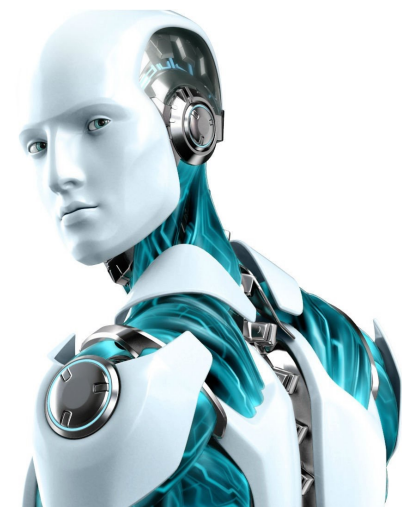


Embedded Law

Embedded Law



Apple WATCH



The Cop in the Backseat



The Cop in the Cockpit



The Cop in Nature

What is that flower?



Can I pick it?

You are in Massachusetts

Picking wild orchids is prohibited in Massachusetts

[Chapter 266 of the General Laws, Section 116A, 1935]

Technology-Enabled Law

Technology-Enabled Law

Enforceable vs. unenforceable regulations

Technology enhances enforceability

From blanket to conditional regulations

From human to automated enforcement

Speed Limit Examples Follow

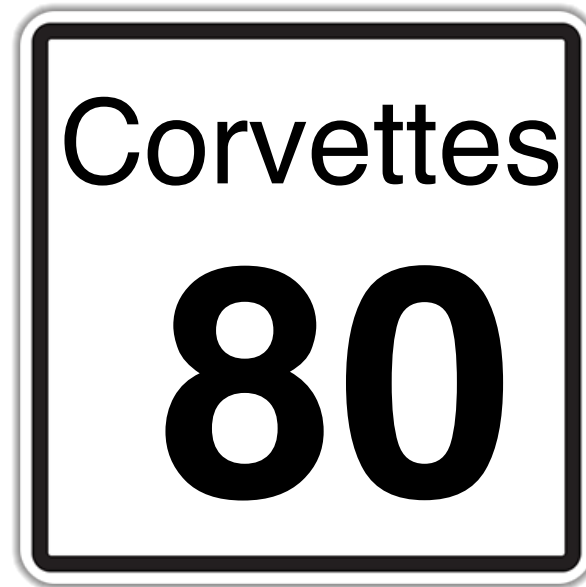
Lane-Specific Speed Limits



Time-Specific Speed Limits



Vehicle-Specific Speed Limits?



Ability-Based Speed Limits?

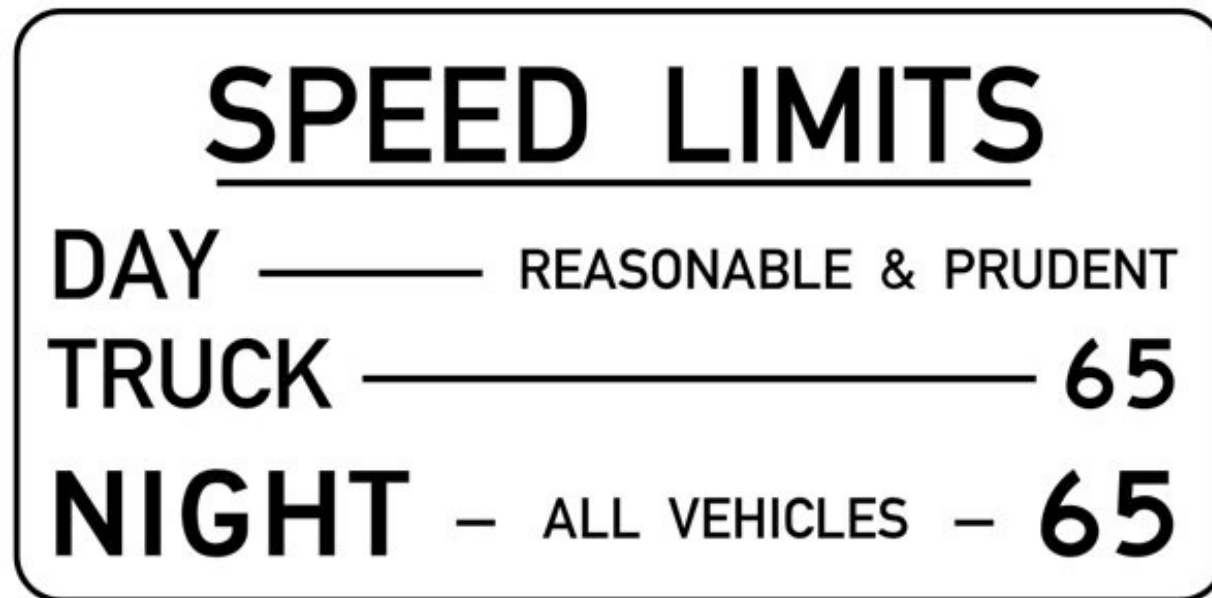
Age 20 - 60

Eyesight 20-20

BP 125/80

60

Montana-Specific Speed Limits!



SCHOOL

SPEED
LIMIT
25

6:49-7:15 AM
7:52-8:22 AM
8:37-9:07 AM
2:03-2:33 PM
3:04-3:34 PM
3:59-4:29 PM

SCHOOL DAYS
ONLY



50
km/h

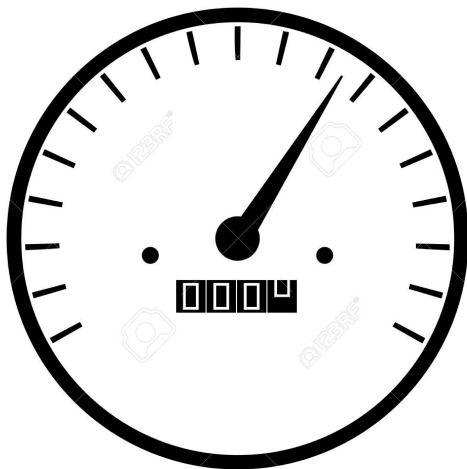
60
km/h

N-76
0157

Automated Enforcement?



Automatic Reporting and Billing



Speed



DMV bill



Insurance bill

Issues

Desirability

Safety

Efficiency

Concerns

Equity - discrimination on the basis of age

Privacy vs automatic DMV reporting

Feasibility

“No way, no how!”

“It’ll never happen.”

Conclusion

Power to the People!

Legal Empowerment Through Information Technology

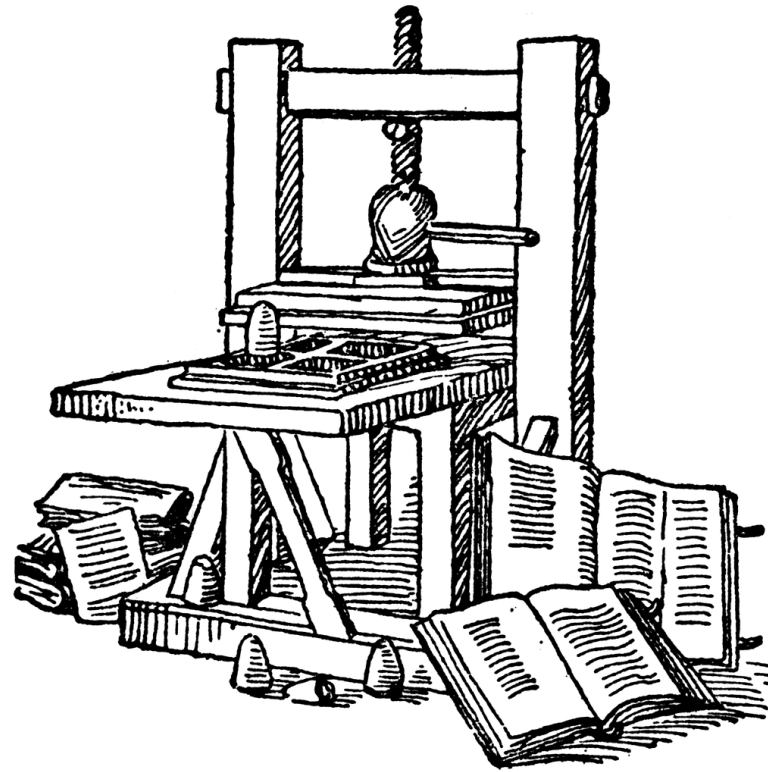


Law Without Lawyers!!

Evolution of Legal Technologies



The Code of Hammurabi





CODEx

The Stanford Center for Legal Informatics

Legal Empowerment through Information Technology

This course

Schedule

Apr 1 Introduction

8 Modeling the World

15 View Definitions and Legality of Situations

22 Action Definitions and Legality of Behavior

29 Comlaw Applications (guest speakers)

May 6 Comlaw Applications (guest speakers)

13 Computable Contracts (guest speakers)

20 Regulatory Implications (guest speakers)

27 Legal Informatics

Theory &
Technology

Applications

Extensions &
Implications

Mathematical Background

Sets

$$\{a, b, c\} \cup \{b, c, d\} = \{a, b, c, d\}$$

$$a \in \{a, b, c\}$$

$$\{a, b, c\} \subseteq \{a, b, c, d\}$$

Functions and Relations

$$\textit{father}(\textit{bob}) = \textit{art}$$

$$\textit{likes}(\textit{john}, \textit{sally})$$

Comfort with use of simple mathematical notation

Grades

Numerical Score

- 5% for Assignment 1
- 10% for each of Assignments 2, 3, 4
- 5% for each of Assignments 5, 6
- 15% for class attendance / participation
- 40% for term project / paper (details to follow)

Discretionary Extra Credit

Added to score before determining Reported Grade

Reported Grade

Based on numerical score (see above)

No curve - independent of number of students

Satisfactory = 70% and above



Computational Law

Spring 2020-2021

CS 204
Law 4019

Lessons

Epilog

Worksheets

Corpus Legis

Piazza

Computational Law is an innovative approach to legal informatics concerned with the representation of regulations in computable form. From a practical perspective, Computational Law is important as the basis for computer systems capable of performing useful legal calculations, such as compliance checking, legal planning, and regulatory analysis. In this course, we look at the theory of Computational Law, we review relevant technology and applications, we discuss the prospects and problems of Computational Law, and we examine its philosophical and legal implications. Work in the course consists of reading, class discussion, and practical exercises. The course is cross-listed with Law 4019.

Michael Genesereth

Email: genesereth@stanford.edu

Office: Gates 220

Office Hours: Wed 3-4

Roland Vogl

Email: rvogl@lawstanford.edu

Office: Crown Quadrangle

Office Hours:

Comments and complaints to genesereth@stanford.edu.



Computational Law

*Law
without
Lawyers*

Lessons

Color Code

Black - Lectures

Blue - Readings

Green - Exercises

Red - Assignments

Grey - Comment

Lesson 1 - Computational Law (Genesereth)

[Lecture 1 - Introduction](#)

[Example - Nine Board Tic Tac Toe](#)

[Example - Gates](#)

[Example - Portico](#)

[Reading 1.1 - Computational Law](#)

[Reading 1.2 - The British Nationality Act as a Logic Program](#)

[Reading 1.3 - The Rule of Law as a Law of Rules](#)

[Reading 1.4 - Computer Elected Governor of California](#)

[Assignment 1.1 - Nine Board Tic Tac Toe](#)

[Assignment 1.2 - Gates](#)

[Assignment 1.3 - Portico](#)



Introduction to Logic Programming

*What
versus
How*

[Lessons](#)

[Sierra](#)

[Tools](#)

[Examples](#)

[Zoom](#)

[Piazza](#)

The following syllabus lists all of the materials of the course. Note that there are interactive exercises at the ends of the chapters in the course textbook. (Click on the exercise numbers to go to the exercise pages.) These exercises are an essential part of the course, and you will benefit from tackling them. Some are easier than others, but you should attempt them all. Do the exercises! Do The Exercises!! DO THE EXERCISES!!!

Color Code

Black - Lecture Slides

[Blue - Readings](#)

[Red - Assignments](#)

Grey - Comment

Introduction (Week 1)

[Lecture 1 - Introduction](#)

[Lecture 2 - Datasets](#)

[Chapter 1 - Introduction](#)

[Chapter 2 - Datasets](#)

[Programs With Common Sense](#)

[Logic Programming](#)

[Assignment 1.1 - Datasets in Sierra](#)

[Assignment 1.2 - Game State](#)

[Assignment 1.3 - Triples](#)

[Project](#)

Textbook

Series ISSN: 1939-4608

SYNTHESIS LECTURES ON ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

Series Editors: Ronald J. Brachman, *Jacobs Technion-Cornell Institute at Cornell Tech*
Francesca Rossi, *AI Ethics Global Leader, IBM Research AI*
Peter Stone, *University of Texas at Austin*

Introduction to Logic Programming

Michael Genesereth, *Stanford University*
Vinay K. Chaudhri, *Stanford University*

“This is a book for the 21st century: presenting an elegant and innovative perspective on logic programming. Unlike other texts, it takes datasets as a fundamental notion, thereby bridging the gap between programming languages and knowledge representation languages; and it treats updates on an equal footing with datasets, leading to a sound and practical treatment of action and change.” – *Bob Kowalski, Professor Emeritus, Imperial College London*

“In a world where Deep Learning and Python are the talk of the day, this book is a remarkable development. It introduces the reader to the fundamentals of traditional Logic Programming and makes clear the benefits of using the technology to create runnable specifications for complex systems.” – *Son Cao Tran, Professor in Computer Science, New Mexico State University*

“Excellent introduction to the fundamentals of Logic Programming. The book is well-written and well-structured. Concepts are explained clearly and the gradually increasing complexity of exercises makes it so that one can understand easy notions quickly before moving on to more difficult ideas.” – *George Younger, student, Stanford University*

ABOUT SYNTHESIS

This volume is a printed version of a work that appears in the *Synthesis Digital Library of Engineering and Computer Science*. Synthesis books provide concise, original presentations of important research and development topics, published quickly, in digital and print formats.



MORGAN & CLAYPOOL PUBLISHERS
store.morganclaypool.com



GENESERETH • CHAUDHRI

INTRODUCTION TO LOGIC PROGRAMMING

MORGAN & CLAYPOOL



MORGAN & CLAYPOOL PUBLISHERS

Introduction to Logic Programming

Michael Genesereth
Vinay K. Chaudhri

SYNTHESIS LECTURES ON ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

Ronald J. Brachman, Francesca Rossi, and Peter Stone, *Series Editors*

Assignments

Readings

[Reading 1.1 - Computational Law](#)

[Reading 1.2 - The British Nationality Act as a Logic Program](#)

[Reading 1.3 - Cabaret: rule interpretation in a hybrid architecture](#)

[Reading 1.4 - The Rule of Law as a Law of Rules](#)

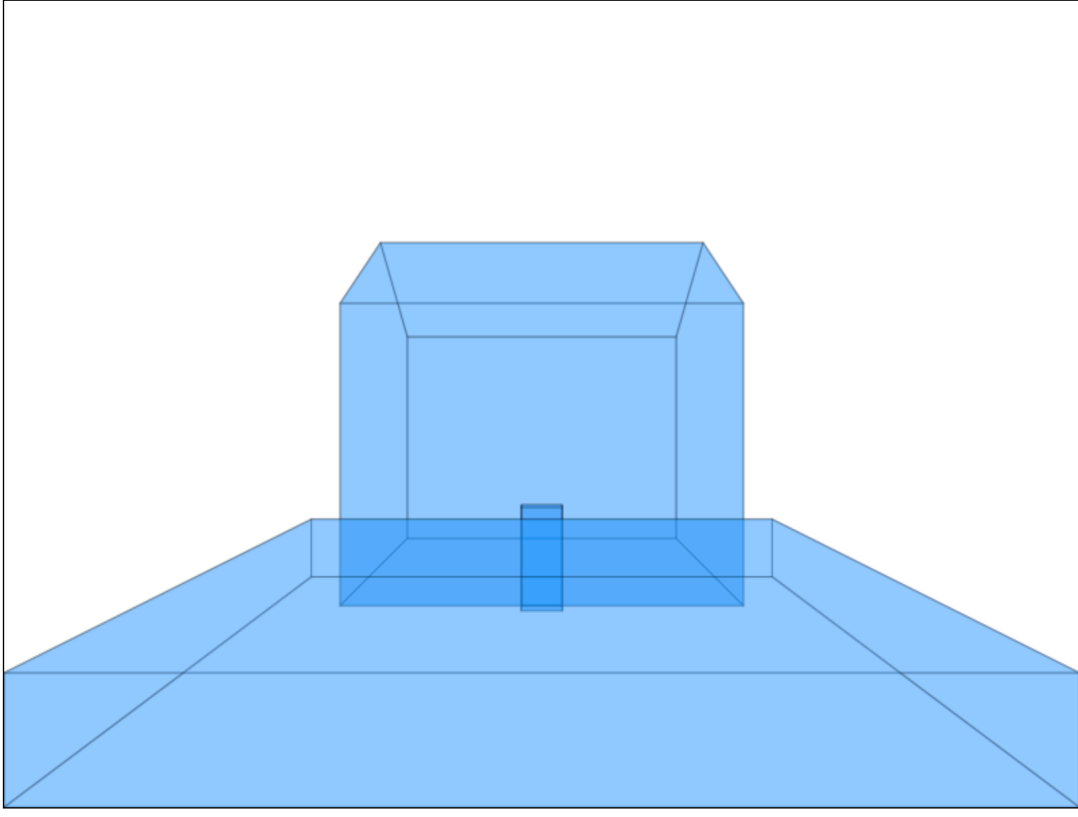
[Reading 1.5 - Computer Elected Governor of California](#)

Assignment - Portico

Not Secure — complaw.stanford.edu

Portico

Use sliders to adjust view. Click and drag to move building. Click Larger, Smaller, Taller, Shorter to adjust size.



Larger Smaller Turn Stop Taller Shorter

Item	Data
Zone	R-1

Standard	Actual	Allowed	Status
Footprint	160000	168000	✓

Item	Min	Max
Home x	200	600

<http://complaw.stanford.edu/chapters/portico.html>

Assignment - Academic Program Sheet

DEPARTMENT OF COMPUTER SCIENCE
MSCS Program Sheet (2010-11)

Artificial Intelligence Primary Specialization

Name: [Charles Parnell Naut](#) Advisor: _____ Proposed date for degree conferral: _____ Date: 10/8/2010
Student ID #: _____ Email: cnaut@stanford.edu _____ _____ HCP? Coterm?

GENERAL INSTRUCTIONS

Before the end of your first quarter, you should complete the following steps. Detailed instructions are included in the **Guide to the MSCS Program Sheet** in your orientation packet (an online version is available at cs.stanford.edu/degrees/mscs/programsheets/):

- Complete this program sheet by filling in the number, name and units of each course you intend to use for your degree.
- Create a course schedule showing the year and quarter in which you intend to take each course in your program sheet.
- Meet with your advisor and secure the necessary signatures on the program sheet.

FOUNDATIONS REQUIREMENT

You must satisfy the requirements listed in each of the following areas; all courses taken elsewhere must be approved by your adviser on a foundation course waiver form. Required documents for waiving a course include course descriptions, syllabi, and textbook lists. These document can be organized here: cs.stanford.edu/degrees/mscs/waivers/. Do not enter anything in the "Units" column for courses taken elsewhere.

Note: If you are amending an old program sheet, enter "**on file**" in the approval column for courses that have already been approved.

Required:	Equivalent elsewhere (course number/title/institution)	Approval	Grade	Units
Logic, Automata and Complexity (<input checked="" type="checkbox"/> CS 103)	_____	_____	_____	4
Probability (<input type="checkbox"/> CS 109, <input type="checkbox"/> STATS 116, <input type="checkbox"/> CME 106, or <input type="checkbox"/> MS&E 220)	_____	_____	_____	_____
Algorithmic Analysis (<input checked="" type="checkbox"/> CS 161)	_____	_____	_____	5
Computer Organization and Systems (<input checked="" type="checkbox"/> CS 107)	_____	_____	_____	5
Principles of Computer Systems (<input checked="" type="checkbox"/> CS 110)	_____	_____	_____	5

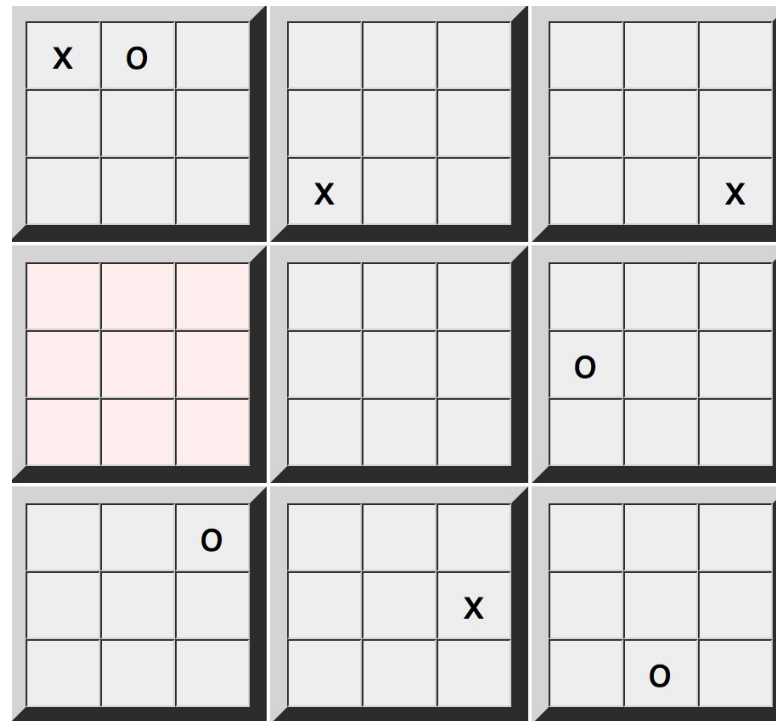
TOTAL UNITS USED TO SATISFY FOUNDATIONS REQUIREMENT: **10**

Note: This total may not exceed 10 units.

✖ 7 Requirements Left Total Units: 10 Status: Draft

<http://complaw.stanford.edu/chapters/program.html>

Assignment - Nineboard Tic Tac Toe



<http://complaw.stanford.edu/chapters/nineboard.html>

Assignment - Teaming

Form a team and choose a name

Composition

3 people each (2 or 4 okay with *good* reason)

Sample Names:

Mighty Bourgeoisie

Michael Genesereth

Red Hot Chili Peppers



CODEx

The Stanford Center for Legal Informatics

Legal Empowerment through Information Technology



CODEx

The Stanford Center for Legal Informatics

Legal Empowerment through Information Technology