

CS 6100 - Foundations of CS¹

Ganesh Gopalakrishnan

School of Computing
University of Utah

January 2012

¹(Beamer slide package courtesy of Carles Mateu, Dept. d'informatica, Univ. Lleida)

CS 6100

Course Overview

Course Overview

Topics

- ▶ Logic
 - ▶ Propositional
 - ▶ First order (including Dec. Procedures)
- ▶ Operational Semantics
 - ▶ Writing semantic specs
- ▶ Concurrency
 - ▶ Model Checking
 - ▶ Shared memory consistency
- ▶ Projects
 - ▶ Select from a list including concurrency spec, code synthesis, test generation, security, other topics...

Assumed Background

- ▶ “CS 3100 material” (ramp-up, if you are lacking)
- ▶ Programming in some declarative language (Ocaml, Scala, Python, others..; excellent chance to acquire this background)

Propositional Logic

Learning Goals

- ▶ Understand Propositional reasoning
 - ▶ Syntactic and semantic
 - ▶ Know how to use propositional logic
 - ▶ Complexity results (NP-completeness)
- ▶ Study mechanization
 - ▶ BDDs
 - ▶ SAT
 - ▶ Understand DPLL
 - ▶ Study SAT algorithms
 - ▶ Implement your own

First-order Logic

Learning Goals

- ▶ Understand FOL
 - ▶ (Semi-) decidability
 - ▶ FOL + theories
- ▶ Study mechanization
 - ▶ SMT solvers
 - ▶ Study the OpenSMT solver (?)
 - ▶ Use solvers for
 - ▶ test generation
 - ▶ program synthesis
 - ▶ others

Operational Semantics

Learning Goals

- ▶ Understand how to write semantic specs
- ▶ Study mechanization/tools
 - ▶ K and Matching Logic
 - ▶ Study program semantics

Concurrency

Learning Goals

- ▶ Understand basics of model-checking
- ▶ Shared memory consistency models
 - ▶ Use Opsem to write concurrency specs

Motivations to study propositional logic

- ▶ Core to CS
- ▶ Reinforces notion of NPC
- ▶ Appreciate power (modeling, solving)
- ▶ Basic engine that supports fast reasoning of more expressive languages (e.g., FOL)

Bradley/Manna topics in Prop. Logic

- ▶ Syntax (know BNF/CFG)
- ▶ Semantic interpretation (computing truth-values inductively)
- ▶ Satisfiability (exists I) and validity (satisfied by all I)
- ▶ Truth-tables
- ▶ Alternative to truth-tables (BDD, see http://en.wikipedia.org/wiki/Binary_decision_diagram, and esp. Knuth's video linked in there.)
- ▶ Validity checking based on semantic arguments ("truth-table method")
- ▶ Using proof-rules, and showing that the negated formula results in a contradiction.
- ▶ Equivalence and implication

topics in Prop. Logic (contd.)

- ▶ Substitution of equivalent formulae
- ▶ Normal forms: NNF, DNF, CNF
- ▶ Exponentiality of CNF DNF inter-conversions
- ▶ Conversions avoiding formula blow-up
- ▶ DPLL
- ▶ Tools: GRASP, Minisat