Learning Targets

Collection for Winter Term 2014/15

Georg Hofferek
IAIK – Graz University of Technology
georg.hofferek@iaik.tugraz.at
Introduction & Motivation

- List and explain uses of logic
  - (in computer science)

- Motivate the need for logic (in computer science) by examples

- Name and explain the “classical questions” in logic
Syntax

- Explain syntax of propositional formulas
  - Based on examples

- Name elements/symbols/connectives of propositional formulas

- Draw parse tree of propositional formulas
Semantics

- Explain semantics of prop. formula
  - Based on a model
- Construct and explain truth table of prop. formula
- Decide validity, satisfiability, semantic entailment/equivalence of prop. formula(s)
  - Using truth tables
- Explain validity, satisfiability, semantic entailment/equivalence
  - Using examples
- Model declarative sentences as prop. formula
  - As detailed as possible
Applications

- Name and explain examples for usage of prop. logic to solve problems

- Solve suitable problems by using prop. logic
  - Reduction to “Classical questions”
Natural Deduction

- Explain natural deduction and its rules
  - Based on examples

- Do a deduction proof or find a counterexample for a given sequent

- Check (or find errors in) a given deduction proof

- Explain “soundness” and “completeness”
  - Of natural deduction for propositional logic
Combinational Equivalence Checking

- Explain the relation between...
  - Satisfiability, validity, entailment, equivalence
  - And how they can be reduced to each other

- Explain “equisatisfiability”
  - Based on an example

- Explain and compute normal forms
  - Of propositional formulas

- Explain & perform Tseitin’s encoding
  - On propositional formulas

- Check equivalence of combinational circuits
  - Using Tseitin’s encoding and a SAT solver
SAT Solving

- Based on example(s), explain
  - SAT Solver
  - DPLL Algorithm
  - Boolean Constraint Propagation
  - Pure Literals
  - Clause Learning
  - Resolution
  - Refutation Proofs

- Use the above tools to proof satisfiability or unsatisfiability of a formula in CNF
  - If satisfiable: Give satisfying model
  - If unsatisfiable: Give refutation proof
Symbolic Computations & Interpolation

- Symbolically encode sets
  - In particular: States, Edges of Graphs

- Perform set operations
  - on symbolically encoded sets

- Explain Craig interpolants
  - and their three main properties

- Compute a Craig interpolant
  - based on a resolution proof,
  - using McMillan’s rules
Binary Decision Diagrams

- Explain BDDs (and components) and their properties, advantages, and disadvantages
  - Based on example

- Determine the function of a given BDD

- Construct a BDD for a given function
  - By computing cofactors
  - Using a given variable order

- Convert a BDD into a multiplexer circuit
Predicate Logic

- Explain syntax of predicate logic
  - Based on examples
- Model (natural, declarative) sentences with predicate logic
- Explain Models of predicate logic
  - Say what they consist of
  - Give examples
- Explain the semantics of predicate logic
  - Based on examples of formulas and models
- Compute the semantics of a formula in predicate logic
  - For a given model
- Explain satisfiability and validity for predicate logic
Natural Deduction for Predicate Logic

- Perform substitution
  - In predicate logic formulas
- Explain the notion of “free for”
  - In context of substitution
  - Give examples
- Explain the predicate-logic-specific rules of natural deduction
  - Using examples
  - Check given “proofs” for correctness
- Proof sequents in predicate logic, or show that they are invalid
  - Using natural deduction proofs, or counterexamples
Real Proofs

- Apply natural deduction to practical problems

- Explain the deductive structure in “everyday” proofs
Theories in Predicate Logic

- Explain what a “Theory in Predicate Logic” is
  - Based on examples
  - State Axioms of $\mathcal{T}_E$ and $\mathcal{T}_{UE}$

- Explain the meaning of “Satisfiability Modulo Theories”
  - Based on examples

- Explain the Concept of Eager Encoding
  - Apply it to Formulas in $\mathcal{T}_{UE}$ Using Ackermann’s Reduction and the Graph-based Reduction

- Explain the Concept of Lazy Encoding
  - Apply it to Formulas in $\mathcal{T}_{UE}$ Using Congruence Closure

- Explain DPLL(T) and its advantages over Eager/Lazy Encoding
Decidability

- Explain what a *decision problem* is
  - Give examples (decidable & undecidable ones)
- Explain *(semi-)* decidability
- Sketch proof of undecidability of predicate logic
  - Using reduction of HALT problem
- Explain relation between problem reduction and decidability
- Explain Gödel’s Incompleteness Theorem
  - Sketch proof