BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI
INSTRUCTION DIVISION
FIRST SEMESTER 2013-2014
Course Handout Part II

In addition to part-I (General Handout for all courses appended to the time table) this portion gives further specific details regarding the course

Course No. : **CS / IS F214**
Course Title **: Logic in Computer Science**

Instructor-in-Charge**: Shan Sundar Balasubramaniam (email: sundarb)**
Instructors:
Vishal Gupta (email: vishalgupta)
Vimal S.P. (email: vimalsp)
Avinash Gautam (email: avinash)
Mayuri Digalwar (email: mayuri)

Course Website: **http://csis/faculty/sundarb/courses/lics**

1. Scope and Objective:

The objective of this course is to introduce the formal study of Logic for Computer Science undergraduates. Within this context the course covers first order propositional and predicate logics as well as temporal logics – LTL and CTL. It also covers some applications in modeling and reasoning about programs - in particular, model checking based on LTL and CTL as well as program verification using Floyd-Hoare logic. Natural Deduction as a proof system for propositional and predicate logics is covered. Soundness and Correctness proofs are also covered but only for propositional logic. The relationship between formal logic and pragmatics of computing is highlighted via a few specialized topics: the satisfiability problem in propositional logic and the challenges in solving it efficiently, Horn-clause problem solving using Prolog programming, and model checking tools.

2. Text Book:

T1: Michael Huth and Mark Ryan. Logic in Computer Science – Modelling and Reasoning about Systems. Cambridge University Press. 2nd Edition. 2004.

Birla Institute of Technology & Science, Pilani
Pilani Campus, Vidya Vihar
Pilani 333031, Rajasthan, India
Tel: +91 1596 245073
Fax: +91 1596 244183
Web: www.pilani.bits-pilani.ac.in

3. Course Plan:

3.a. Modules

Module #	Topics
I	**Introduction and Review of Logics and Proofs. Overview of the interplay of Logic and Computing.**
II	**Propositional Logic: Natural Deduction, Syntax and Semantics, Soundness and Completeness, Satisfiability.**
III	**Predicate Logic: Syntax and Semantics, Logic Programming, Natural Deduction, Software Modeling**
IV	**Temporal Logic: LTL, CTL, and Model Checking**
V	**Program Verification: Floyd-Hoare Logic – Pre-conditions, Post-conditions, and Loop Invariants.**

3. b. Lecture Schedule:

LEGEND:

Ln – Lecture n	Tun – Tutorial n
An – Assignment n	SL – Self Learning

End of LEGEND

Lecture / Tutorial #	Module #	Topics	Learning Outcome(s) [The student will be able to:]	Reading
L1	I.a	Why study Logic?	• state a few reasons to study logics and proofs	-
L1	I.b	A broad and selective history of Logic and Proofs in the last century: Hilbert's Program, Russell's efforts, and Brouwer's intuitionism.	• understand typical issues and debates in formalizing proofs (and mathematical arguments in general)	-

Birla Institute of Technology & Science, Pilani
Pilani Campus, Vidya Vihar
Pilani 333031, Rajasthan, India

Tel: +91 1596 245073
Fax: +91 1596 244183
Web: www.pilani.bits-pilani.ac.in

Tu1	I. b	Russell's Paradox: Definitions in Set Theory. Non-Constructive Proofs: Use of the Law of Excluded Middle	• explain the difference between a constructive and a non-constructive proof	-
Tu2	I.c	Mathematical Induction: Examples, Structure of the proofs.	• explain the structure of a proof using induction and • handle typical issues in applying induction	-
L2	I.d	Logic and Computing: Godel's Incompleteness Results and their impact on the formalization of mathematics	• state soundness and completeness requirements in the context of proof systems and • understand – at a high level – the implications of Godel's incompleteness results	-
L2	I.e	Godel, Church, and Turing: Computability – Recursive Functions, Lambda Calculus, and, Turing Machines; Church-Turing Thesis.	• Informally define the notion of "computability" • state the equivalence of schemes / mechanisms for computability	-
L2	I.f	Time Complexity, Complexity Classes, Is P = NP?	• define complexity classes P and NP	-
Tu3	I.g	Boolean Logic and Boolean operations.	• apply combinatorial arguments on functions as well as • use Boolean algebraic operations	-
L3	I.h	Boolean Satisfiability (SAT) and its membership in NP. Horn-Clause Satisfiability and its membership in P.	• (by hand) verify satisfiability of Boolean expressions and Horn-style expressions.	-
L3	I.i	Logic and Programming: Logic Programming and Prolog.	• state the relation between logic and programming at a high level	-

Birla Institute of Technology & Science, Pilani
Pilani Campus, Vidya Vihar
Pilani 333031, Rajasthan, India

Tel: +91 1596 245073
Fax: +91 1596 244183
Web: www.pilani.bits-pilani.ac.in

L4	II.a	Propositional Logic: Natural Deduction: Conjunction Rules	• apply proof rules involving conjunction	T1 Sec. 1.2.1
L4	II.b	Propositional Logic: Natural Deduction: Double Negation	• apply proof rules involving double negation	T1 Sec. 1.2.1
L4	II.c	Propositional Logic: Natural Deduction: Implication	• apply proof rules involving implication	T1 Sec. 1.2.1
Tu4	II.d	Propositional Logic: Natural Deduction: Proofs: Disjunction Rules	• apply proof rules involving disjunction	T1 Sec. 1.2.1
Tu4	II.e	Propositional Logic: Natural Deduction: Proofs - Examples	• prove sentences in propositional logic using natural deduction	T1 Sec. 1.2.1
Tu5	II.f	Propositional Logic: Natural Deduction: Proofs - Examples	• prove sentences in propositional logic using natural deduction	T1 Sec. 1.2.1
L5	II.g	Propositional Logic: Natural Deduction: Negation Rules	• apply proof rules involving negation	T1 Sec. 1.2.1
L5	II.h	Propositional Logic: Natural Deduction: Derived Rules: *Modus Tollens*	• apply *modus tollens*	T1 Sec. 1.2.2
L5	II.i	Propositional Logic: Natural Deduction: Derived Rules: Proof By Contradiction	• apply proof by contradiction	T1 Sec. 1.2.2
L5	II.j	Propositional Logic: Natural Deduction: Derived Rules: Law of Excluded Middle	• apply LEM	T1 Sec. 1.2.2
L6	II.k	Syntax: Context Free Grammars: Backus-Naur-Form: Notation and Examples	• define CFGs for simple constructs	T1 Sec. 1.3
L6	II.l	Context Free Grammars: Parse Trees: Examples	• illustrate and explain the internal structure of context-free	T1 Sec. 1.3

Birla Institute of Technology & Science, Pilani
Pilani Campus, Vidya Vihar
Pilani 333031, Rajasthan, India

Tel: +91 1596 245073
Fax: +91 1596 244183
Web: www.pilani.bits-pilani.ac.in

			constructs	
L6	II.m	Propositional Logic: Syntax: Well-formed-formulas and grammar.	• state the formal syntax of propositional logic	T1 Sec. 1.3
Tu6	II.n	Grammars, Syntax Trees, and Structural Induction - Examples	• illustrate the relation between languages and inductive proofs • explain structural induction as a proof technique	T1 Sec. 1.4.2
Tu7	II.o	Structural Induction - Examples	• write proofs using structural induction	T1 Sec. 1.4.2
SL	II.p	Propositional Logic: Semantics	• interpret sentences in propositional logic	T1 Sec. 1.4.1
L7	II.q	Propositional Logic : Soundness	• state soundness arguments in general and soundness arguments for logics in particular	T1 Sec. 1.4.3
L7	II.r	Propositional Logic: Soundness Proof	• argue that proofs in propositional logic are sound	T1 Sec. 1.4.3
L8	II.s	Propositional Logic: Completeness	• state completeness arguments in general and completeness arguments for logics in particular	T1 Sec. 1.4.4
L8	II.t	Propositional Logic: Completeness Proof Overview	• state completeness arguments for propositional logic	T1 Sec. 1.4.4
L8	II.u	Propositional Logic: Completeness Proof	• argue that propositional logic is complete	T1 Sec. 1.4.4
Tu8	II.v	Propositional Logic: Semantic Equivalence	• argue the equivalence – or lack of it – of statements in propositional logic	T1 Sec. 1.5.1

Birla Institute of Technology & Science, Pilani
Pilani Campus, Vidya Vihar
Pilani 333031, Rajasthan, India

Tel: +91 1596 245073
Fax: +91 1596 244183
Web: www.pilani.bits-pilani.ac.in

Tu 8:	II.w	Propositional Logic: Conjunctive Normal Form	• recognize propositional logic formulas in CNF • rewrite propositional logic formulas in CNF	T1 Sec. 1.5.2
Tu 8:	II.x	Propositional Logic: Validity	• verify whether a propositional logic formula is valid or not	T1 Sec. 1.5.1
Tu 9:	II.y	Propositional Logic: Satisifiability	• verify whether a propositional logic formula is satisfiable or not	T1 Sec. 1.5.1
Tu 9:	II.z	Propositional Logic: Normal Forms, Validity, and Satisfiability	• relate validity and satisfiability arguments to CNF	T1 Sec. 1.5.2
L9:	II.aa	Propositional Logic: CNF and Validity	• verify whether a propositional logic formula in CNF is valid or not	T1 Sec. 1.5.2
L9:	II.ab	Propositional Logic: Algorithm for Validity	• write a program to verify validity of a formula in CNF	T1 Sec. 1.5.2
L10	II.ac	Propositional Logic: Horn Clauses and Horn Formulas	• distinguish Horn formulas from general formulas in propositional logic	T1 Sec. 1.5.3
L10	II.ad	Propositional Logic: Satisifiability of Horn Formulas	• verify whether a Horn formula is satisfiable or not	T1 Sec. 1.5.3
L10	II.ae	Propositional Logic: Algorithm for Satisfiability of Horn Formulas	• write a program to verify validity of a Horn formula	T1 Sec. 1.5.3
Tu10	II.af	Propositional Logic: CNF and Horn Formulas	• illustrate and explain the relation between CNF and Horn clause form	T1 Sec. 1.5.3
Tu10	II.ag	Propositional Logic: Simplified SAT Solver	• state the intricacies of a SAT solver	T1 Sec. 1.6
SL	II.ai	Propositional Logic: SAT Solvers: Correctness vs. Complexity	• implement a simple SAT solver	T1 Sec. 1.6

Birla Institute of Technology & Science, Pilani
Pilani Campus, Vidya Vihar
Pilani 333031, Rajasthan, India

Tel: +91 1596 245073
Fax: +91 1596 244183
Web: www.pilani.bits-pilani.ac.in

Tu11	III.a	Expressiveness of Propositional Logic - Limitations	• illustrate statements and arguments that cannot be expressed in propositional logic	T1 Sec. 2.1
L11	III.b	Predicate Logic: Features	• write formulas in predicate logic	T1 Sec. 2.2
L11	III.c	Predicate Logic: Soundness and Completeness	• state that predicate logic is sound and complete	T1 Sec. 2.2
L11	III.d	Predicate Logic: Syntax: Terms and Formulas	• define the formal syntax of predicate logic	T1 Sec. 2.2.1 & 2.2.2
L12	III.e	Predicate Logic: Syntax: Free and Bound Variables	• distinguish between free variables and bound variables in a predicate logic formula	T1 Sec. 2.2.3
L12	III.f	Predicate Logic: Syntax: Substitution	• apply substitution of terms on a predicate logic formula	T1 Sec. 2.2.4
L12	III.g	Predicate Logic: Horn Clauses	• recognize Horn clauses	Notes
Tu12	III.h	Logic Programming: Prolog: Facts and Queries	• specify simple facts and queries on facts using Prolog	Notes
Tu12	III.i	Logic Programming: Prolog: Rules and Queries	• specify simple formulas in predicate logic as rules and queries in Prolog	Notes
Tu13	III.j	Logic Programming: Prolog: Unification	• apply unification on Prolog terms	Notes
Tu13	III.j	Logic Programming: Prolog: Search and Backtracking	• explain how rule search works in Prolog	Notes
A1 (Start)	**III.k**	**Logic Programming: Problem Solving in Prolog**	-	-
L13	III.l	Predicate Logic: Natural Deduction: Rules for Equality	• apply rules for equality in proofs of predicate logic formulas	T1 Sec. 2.3.1

Birla Institute of Technology & Science, Pilani
Pilani Campus, Vidya Vihar
Pilani 333031, Rajasthan, India

Tel: +91 1596 245073
Fax: +91 1596 244183
Web: www.pilani.bits-pilani.ac.in

L13	III.m	Predicate Logic: Natural Deduction: Rules for Universal Quantification	• apply rules for universal quantification in proofs of predicate logic formulas	T1 Sec. 2.3.1
L14	III.n	Predicate Logic: Natural Deduction: Rules for Existential Quantification	• apply rules for existential quantification in proofs of predicate logic formulas	T1 Sec. 2.3.1
L14	III.o	Predicate Logic: Natural Deduction: Proofs	• illustrate and explain the structure of natural deduction proofs of predicate logic formulas	T1 Sec. 2.3.1
Tu14	III.p	Predicate Logic: Quantifier Equivalences	• deduce equivalences of formulas in predicate logic	T1 Sec. 2.3.2
Tu15	III.q	Predicate Logic: Proofs using Natural Deduction	• prove predicate logic formulas using natural deduction	T1 Sec. 2.3
L15	III.r	Predicate Logic: Semantics	• describe the semantics of predicate logic formulas	T1 Sec. 2.4
L15	III.s	Predicate Logic: Semantics: Models	• describe what models are	T1 Sec. 2.4.1
L15	III.t	Predicate Logic: Semantics: Model Checking	• describe the role of models in proofs	T1 Sec. 2.4.1
L16	III.u	Predicate Logic: Semantics: Semantic Entailment	• illustrate and explain semantic entailment in predicate logic	T1 Sec. 2.4.2
L16	III.v	Predicate Logic: Semantics: Semantics of Equality	• illustrate and explain the meaning of equality in predicate logic	T1 Sec. 2.4.3

Birla Institute of Technology & Science, Pilani
Pilani Campus, Vidya Vihar
Pilani 333031, Rajasthan, India

Tel: +91 1596 245073
Fax: +91 1596 244183
Web: www.pilani.bits-pilani.ac.in

L16	III.w	Predicate Logic: Semantics: Validity and Undecidability of Validity	• define validity of predicate logic formulas • state the undecidability of "validity" in predicate logic	T1 Sec. 2.5
Tu16	III.x	Predicate Logic: Expressiveness: Inexpressible Properties: Example(s).	• provide examples of statements that cannot be expressed in first order predicate logic	T1 Sec. 2.6
Tu16	III.y	Existential Second Order Logic: Expressiveness: Examples(s)	• provide examples of statements that require existential quantification over predicates	T1 Sec. 2.6.1
Tu17	III.z	Universal Second Order Logic: Expressiveness: Example(s)	• provide examples of statements that require universal quantification over predicates	T1 Sec. 2.6.2
A1 (end)	-	-	• specify problems and solutions as Prolog programs	-
L17	III.aa	Software Modeling: State Machines	• illustrate and explain specification using state machines	T1 Sec. 2.7.1
L17	III.ab	Software Modeling: Model Checking	• describe model checking	T1 Sec. 2.7.1
SL	III.ad	Model Checking using State Machine Models	• illustrate and explain model checking using state machine models	T1 Sec. 2.7.2 & 2.7.3
Tu19	IV.a	Truth and Time.	• provide examples where "time" needs to be specified explicitly in logical formulas	T1 Sec. 3.1
Tu19	IV.b	Modeling Time – Approaches and Examples	• provide examples where different notions of "time" are explicitly used in logical formulas	T1 Sec. 3.1

Birla Institute of Technology & Science, Pilani
Pilani Campus, Vidya Vihar
Pilani 333031, Rajasthan, India

Tel: +91 1596 245073
Fax: +91 1596 244183
Web: www.pilani.bits-pilani.ac.in

L18	IV.c	Linear-time Temporal Logic	• describe LTL	T1 Sec. 3.2
L18	IV.d	LTL: Syntax: Formulas and Parse Trees	• provide examples of formulas in LTL • illustrate internal structure of formulas in LTL	T1 Sec. 3.2.1
L19	IV.e	LTL: Semantics: Transitions	• interpret LTL formulas	T1 Sec. 3.2.2
L19	IV.f	LTL: Semantics: Paths	• interpret LTL formulas	T1 Sec. 3.2.2
Tu20	IV.g	LTL: Specifications: Examples	• write LTL specifications for complex scenarios	T1 Sec. 3.2.3
Tu21	IV.h	LTL: Equivalences	• argue equivalence of formulas in LTL	T1 Sec. 3.2.4
Tu21	IV.i	LTL: Adequacy vs. Orthogonality of connectives	• argue adequacy of constructs in LTL	T1 Sec. 3.2.5
A2 (start)	**IV.j**	**Model Checking using tools**	-	T1 Sec. 3.3.1 – 3.3.3
L20	IV.k	Model Checking: Case Study: Problem Statement	• define the problem of model checking for a particular case	T1 Sec. 3.3.4 & 3.3.5
L20	IV.l	Model Checking: Case Study: Modeling approach and Issues	• explain intricacies and issues in model checking	T1 Sec. 3.3.4 & 3.3.5
SL	IV.m	Model Checking: Tool(s)	• use model checking tool(s)	T1 Sec. 3.3.1 – 3.3.6

Birla Institute of Technology & Science, Pilani
Pilani Campus, Vidya Vihar
Pilani 333031, Rajasthan, India

Tel: +91 1596 245073
Fax: +91 1596 244183
Web: www.pilani.bits-pilani.ac.in

L21	IV.n	Branching Time Logic: Computation Tree Logic	• describe CTL	T1 Sec. 3.4
L21	IV.o	CTL: Syntax: Formulas and Parse Trees	• provide examples of formulas in CTL • illustrate internal structure of formulas in CTL	T1 Sec. 3.4.1
L21	IV.p	CTL: Semantics: Semantic Entailment	• interpret CTL formulas	T1 Sec. 3.4.2
Tu22	IV.q	CTL: Specifications	• write CTL specifications for complex scenarios	T1 Sec. 3.4.3
Tu22	IV.r	LTL and CTL: Expressive Powers	• provide examples of scenarios which cannot be specified in CTL but not in LTL and vice versa	T1 Sec. 3.5
Tu23	IV.s	CTL: Model Checking	• illustrate and explain how model checking works in CTL	T1 Sec. 3.6
L22	IV.t	CTL: Model Checking: Algorithm	• implement a model checker for CTL	T1 Sec. 3.6.1
L23	IV.u	LTL: Model Checking: Algorithm	• implement a model checker for LTL	T1 Sec. 3.6.3
SL	IV.v	LTL: Model Checking: Tool(s)	• use LTL model checking tool(s)	-
Tu24	V.a	Program Verification: Floyd-Hoare Logic: Pre-conditions and Post-Conditions	• describe what pre-conditions and post-conditions are	T1 Sec. 4.2.2
Tu24	V.b	Program Verification: Floyd-Hoare Logic: Assignment Statements and Sequencing	• write pre-conditions and post-conditions for assignment statements • compose pre-conditions and post-conditions over sequential statements	T1 Sec. 4.2.2 & 4.3.1

Birla Institute of Technology & Science, Pilani
Pilani Campus, Vidya Vihar
Pilani 333031, Rajasthan, India

Tel: +91 1596 245073
Fax: +91 1596 244183
Web: www.pilani.bits-pilani.ac.in

Tu24	V.c	Program Verification: Floyd-Hoare Logic: Conditional Statements	• compose pre-conditions and post-conditions over conditional statements	T1 Sec. 4.2.2 & 4.3.1
L24	V.d	Program Verification: Hoare Triples	• illustrate and explain how Floyd-Hoare logic works (over simple programs) • illustrate and explain Hoare triples and their meaning	T1 Sec. 4.2.2
L24	V.e	Program Verification: Partial Correctness and Total Correctness	• distinguish between partial correctness arguments and total correctness arguments	T1 Sec. 4.2.3
A2 (end)	-	-	• **perform model checking of reasonably complex models using tools**	-
L25	V.f	Program Verification: Floyd-Hoare Logic: Loop Invariants	• illustrate and explain what loop invariants are	T1 Sec. 4.2.2 & 4.3.1
L25	V.g	Program Verification: Floyd-Hoare Logic: Verifying correctness of Loops	• write loop invariants given simple loops • provide correctness arguments using loop invariants	T1 Sec. 4.2.2 & 4.3.1
Tu25	V.h	Program Verification: Floyd-Hoare Logic: Proof Rules and Verification Examples	• provide correctness arguments for small programs	T1 Sec. 4.3.1 & 4.3.2
Tu26	V.i	Program Verification: Floyd-Hoare Logic: Case Study	• explain intricacies of proving programs using Floyd-Hoare logic	T1 Sec. 4.3.3
L26	V.j	Program Verification: Limitations	• describe issues in and limitations of proving correctness of programs	-
L26	-	Course Summary	• summarize what was learnt in the course	-

Birla Institute of Technology & Science, Pilani
Pilani Campus, Vidya Vihar
Pilani 333031, Rajasthan, India

Tel: +91 1596 245073
Fax: +91 1596 244183
Web: www.pilani.bits-pilani.ac.in

4. Evaluation

4. a. Evaluation Scheme:

Component	Weight	Date	Remarks
Quizzes (4 out of 5)	4x15M=60M	In Tutorial Sessions	1 session notice (Closed Book)
Assignments (2)	2x21M=42M	2 to 3 weeks one in Sep. and one in Nov.	Take Home (Teams of 2)
Mid-Term Test (90 minutes)	42M	<test_1>	(scheduled centrally) (Open Book)
Comprehensive Exam (120 minutes)	56M	<test_c>	(scheduled centrally) (Open Book)
TOTAL	200M	-	-

4. b. Make-up Policy:

- **No Make-up will be available for Quizzes and Assignments under any condition.**
- **Late submission of assignment will incur a penalty of 25% up to 24 hours and a penalty of 50% up to 48 hours from the deadline.**
- **Prior Permission of the Instructor-in-Charge is usually required to get make-up for the mid-term test.**
- **Prior Permission of Dean Instruction is usually required to get make-up for the comprehensive exam.**
- **A make-up shall be granted only in genuine cases where - *in the Instructor's / Dean's judgment* - the student would be physically unable to appear for the test/exam. Instructor's / Dean's decision in this matter would be final.**

4.c. Fairness Policy:

- **Student teams are expected to work on their own on assignments.**
- **All students are expected to contribute equally within a team. The instructor's assessment regarding the contributions of team members would be final.**
- **Any use of unfair means in quizzes, assignment, or test/exam will be reported to the Unfair means committee and will be subject to the severest penalty possible:**
  - **Unfair means would include copying from other students or from the Web or from other sources of information including electronic devices.**

5. Consultation Hours: See course website.

6. Notices: All notices concerning this course will be displayed on the course website only. If there is a need email would be used on short notice (12 hours) – only BITS Pilani mail would be used.

Birla Institute of Technology & Science, Pilani
Pilani Campus, Vidya Vihar
Pilani 333031, Rajasthan, India

Tel: +91 1596 245073
Fax: +91 1596 244183
Web: www.pilani.bits-pilani.ac.in

Instructor –In- Charge

CS F214 / SS F214

Birla Institute of Technology & Science, Pilani
Pilani Campus, Vidya Vihar
Pilani 333031, Rajasthan, India

Tel: +91 1596 245073
Fax: +91 1596 244183
Web: www.pilani.bits-pilani.ac.in