CE 474 - LOGIC OF COMPUTER SCIENCE

Lecture Note 6

1. Learning Objectives

By the end of this lecture, you should be able to:

- State the Soundness Theorem and Completeness Theorem for propositional logic.
- Understand the high-level ideas behind their proofs.
- Appreciate why these meta-theorems matter in computer-science applications (e.g., verification).
- Engage in a critical discussion of expressiveness vs. decidability trade-offs.

2. Key Definitions

- **Proof System** ($\vdash$): A formal system (e.g., natural deduction) with inference rules to derive formulas.
- **Semantic Entailment** ($\models$): $\Gamma \models \varphi$ means every truth-value assignment that makes all formulas in Γ true also makes φ true.
- **Valid Formula**: A formula φ such that $\models \varphi$ (true under all assignments).
- **Provable Formula**: A formula φ such that $\vdash \varphi$ (derivable without premises).

3. Soundness Theorem

Statement:

$$\text{If } \Gamma \vdash \varphi, \text{ then } \Gamma \models \varphi.$$

Interpretation: Anything you can prove syntactically is also semantically valid.
Why it holds (Proof Sketch):

- **Base Case:** Axioms are tautologies, hence always true.
- **Inductive Step:** Show that natural deduction rules (like $\wedge$-introduction, $\rightarrow$-elimination) preserve truth.

4. Completeness Theorem

Statement:

$$\text{If } \models \varphi, \text{ then } \vdash \varphi.$$

Interpretation: Any semantically valid formula is also provable in our system.
Why it holds (Proof Sketch):

- Extend consistent sets of formulas to *maximal consistent sets*.
- Define a valuation v such that $v(p) = \text{true}$ iff p is in the set.
- Prove that for every formula ψ, ψ is in the set iff $v \models \psi$.
- Use contraposition: if φ is not provable, $\neg\varphi$ can be added consistently. Then construct a model where φ is false.

5. Implications for Computer Science

- **Verification:** Soundness ensures correctness proofs reflect true behavior.
- **Automated Theorem Proving:** Completeness guarantees that valid properties are, in principle, discoverable.
- **Decidability:**
  - Propositional logic is decidable — SAT solvers can determine satisfiability.
  - First-order logic is incomplete (*Gödel*) and undecidable.

6. In-Class Discussion Prompt

Question:

> *Why might we choose a weaker logic (e.g., propositional or monadic first-order logic) instead of full first-order logic in verification tasks?*

Discussion Points:

- Trade-off between expressive power and automated tool support.
- Propositional logic has efficient solvers (SAT/SMT).
- First-order logic can model more, but is harder to reason automatically.

7. Summary & Key Takeaways

- **Soundness:** If provable, then semantically valid.
- **Completeness:** If semantically valid, then provable.
- These two properties establish trust in our formal reasoning tools.
- Practical tools often restrict to decidable fragments (e.g., propositional) for performance.

References

- Huth, M., & Ryan, M. (2004). *Logic in Computer Science: Modelling and Reasoning about Systems*. Cambridge University Press.
- Enderton, H. B. (2001). *A Mathematical Introduction to Logic* (2nd ed.). Academic Press.
- Troelstra, A. S., & Schwichtenberg, H. (2000). *Basic Proof Theory* (2nd ed.). Cambridge University Press.
- Sipser, M. (2012). *Introduction to the Theory of Computation*. Cengage Learning.